

UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

INGENIERÍA DE TELECOMUNICACIÓN



PROYECTO FIN DE CARRERA

***Evaluación de algoritmos supervisados de extracción de
características para clasificación de texturas***

Autor: Guillermo Mariñas Miranda

Tutor: Dr. Jerónimo Arenas García

DICIEMBRE DE 2009

TÍTULO: Evaluación de algoritmos supervisados de extracción de características para clasificación de texturas.

AUTOR: Guillermo Mariñas Miranda.

TUTOR: Jerónimo Arenas García.

La defensa del presente Proyecto Fin de Carrera se realizó el día 22 de diciembre de 2009, siendo calificada por el siguiente tribunal:

PRESIDENTE:

VOCAL:

SECRETARIO:

Habiendo obtenido la calificación de

Presidente

Vocal

Secretario

Leganés, a ____ de _____ de _____

RESUMEN

En este Proyecto Fin de Carrera se plantea el problema de la extracción de características, mediante distintos métodos, en el ámbito de la clasificación de texturas. Dichos métodos consisten en procesar el espectro de las imágenes por un banco de filtros para, a partir de ahí, extraer las características que más información proporcionen para la posterior fase de clasificación.

Concretamente, se compararán dos métodos alternativos; uno de ellos ha sido ya ampliamente usado en clasificación de texturas, y sus prestaciones servirán como referencia; el otro, el cual es nuestro principal objeto de estudio, ha sido aplicado satisfactoriamente en la clasificación de géneros musicales, y se pretende su extrapolación para el problema de la clasificación de texturas. Dichos métodos son, respectivamente:

- Extracción mediante un banco de filtros de Gabor, el cual es fijo, y está basado en el sistema de reconocimiento del cerebro humano.
- Extracción mediante filtros variables, adaptados a la base de datos, obtenidos mediante un método de aprendizaje máquina supervisado denominado POPLS.

Una vez establecido el sistema de clasificación para ambos métodos se evaluarán por separado sus prestaciones: tasa de aciertos y matriz de confusión, para determinar la viabilidad del método supervisado.

ABSTRACT

This Dissertation presents the problem of feature extraction, by means of different methods, inside the scope of texture classification. Such methods consist of processing the images spectrum with a filter bank, in order to extract the most relevant information to achieve good classification rates.

Specifically, two alternative methods will be compared; one of them has already been widely used in texture classification, and its features will serve as reference; the other one, which is our main object of study, has been satisfactorily applied to music genre classification, and our intent is its extrapolation to the field of texture classification. Said methods are, respectively:

- Extraction by means of a Gabor filter bank, which is fixed and based on the human brain recognition system.
- Extraction through variable filters, tuned to the database, which are obtained by means of a supervised machine learning method known as POPLS.

Once the classification system for both methods has been established, their features will be evaluated separately: hit rate and confusion matrix, to determine the viability of the supervised method.

Índice general

Índice general	I
Lista de Figuras	V
Lista de Tablas	IX
Capítulo 1. Introducción a la clasificación de texturas.....	1
1.1. Motivación de la clasificación de patrones	2
1.2. Aplicaciones de la clasificación de patrones	3
1.2.1. Reconocimiento del habla	3
1.2.2. Clasificación de textos.....	4
1.2.3. Reconocimiento de caligrafía	5
1.2.4. Reconocimiento de caras	5
1.3. Un Sistema de clasificación	6
1.3.1. Generalidades del aprendizaje máquina	6
1.3.2. Preprocesado.....	9
1.3.3. Extracción de características	10
1.3.4. Clasificación	13
1.3.5. Conclusiones.....	16
1.4. Clasificación de texturas	17
1.4.1. Métodos de clasificación de texturas.....	18
1.4.2. Las texturas como patrones periódicos.....	19
1.4.3. Aplicaciones de la clasificación de texturas	20
1.4.4. Extracción de características mediante filtros de Gabor	21

1.4.5. Extracción de características mediante POPLS	23
1.5. Objetivo del Proyecto	24
Capítulo 2. Extracción de características mediante filtros de Gabor	27
2.1. Fundamentos de los filtros de Gabor bidimensionales	27
2.1.1. La portadora sinusoidal compleja	28
2.1.2. La envolvente gaussiana	29
2.1.3. La función de Gabor compleja.....	30
2.1.4. Perfil de magnitud mitad.....	32
2.1.5. Filtro de Gabor simplificado y ejemplos	34
2.2. Uso de filtros de Gabor en la clasificación de texturas	37
2.2.1. Búsqueda de similitud de texturas	37
2.2.2. Clasificación de imágenes invariantes a la rotación	40
2.3. Conclusiones.....	46
Capítulo 3. Extracción de características mediante Análisis de Múltiples Variables.....	47
3.1. Análisis de Componentes Principales.....	48
3.1.1. Planteamiento del problema.....	48
3.1.2. Cálculo de las componentes.....	50
3.2. Algoritmos de MVA supervisados	52
3.2.1. Mínimos cuadrados parciales.....	53
3.2.2. PLS ortonormalizado	54
3.2.3. OPLS con restricciones de positividad	56
3.3. Comparación de los distintos métodos	58
3.4. Conclusiones.....	60
Capítulo 4. Experimentos: Clasificación de texturas	63

4.1. Elección de la base de datos	63
4.2. Preprocesado de imágenes	67
4.2.1. Conversión a formato blanco y negro	67
4.2.2 Recorte a 500x500 píxeles.....	68
4.2.3. Transformación al dominio frecuencial.....	70
4.2.4. Promediado del espectro a 10 x 10 puntos	71
4.3. Descripción del clasificador empleado	74
4.4. Extracción de características mediante filtros de Gabor	76
4.4.1. Cobertura espectral	77
4.4.2. Elección de características tras el filtrado	78
4.4.3. Elección del número de filtros.....	79
4.5. Extracción de características mediante POPLS	83
4.5.1. Algoritmo de extracción	83
4.5.2. Análisis del primer filtro obtenido	85
4.5.3. Evolución de las prestaciones con el número de filtros	87
4.6. Comparativa entre ambos métodos	90
4.7. Conclusiones	93
Capítulo 5. Conclusiones y líneas futuras	95
5.1. Conclusiones	95
5.2. Líneas futuras de investigación.....	97
Apéndice A. Conceptos básicos del procesamiento digital de imágenes.....	99
A.1. Representación de imágenes en el dominio de la frecuencia	99
A.2. Filtrado en el dominio de la frecuencia	101
A.2.1. Generalidades	101
A.2.2. Eliminación de la componente continua.....	102

A.2.3. Filtro paso alto	103
A.2.4. Filtro paso bajo.....	106
A.2.5. Filtro de banda rechazada	108
Apéndice B. Filtros de Gabor unidimensionales	111
B.1. Generalidades.....	111
B.2. Eliminación de la componente continua	112
Bibliografía.....	115

Lista de Figuras

Figura 1.1: Sistema de clasificación de árboles. (a) Un conjunto de árboles de muestra para el entrenamiento. (b) Nueva imagen que deberá ser clasificada.	9
Figura 1.2: Envolvente del histograma que representa la intensidad de las hojas de los árboles con que se entrenará la máquina.	11
Figura 1.3: Envolvente del histograma que representa la relación ancho / alto de los árboles para cada una de las muestras de entrenamiento.	11
Figura 1.4: Diagrama de dispersión donde se tienen en cuenta simultáneamente tanto la intensidad como la relación ancho / alto de cada árbol para cada clase....	12
Figura 1.5: Clasificación mediante una frontera lineal entre las dos clases.....	15
Figura 1.6: Clasificación mediante un modelo de frontera más complicado que el de la Figura 1.5.....	15
Figura 1.7: Clasificación mediante una frontera de decisión algo más compleja que en la Fig. 1.5 pero no tan compleja como en la Fig. 1.6 con el objeto de obtener una buena generalización.	16
Figura 1.8: Diferentes tipos de texturas: (a) ladrillos, (b) hierba y (c) cielo.....	20
Figura 1.9: Dos filtros de Gabor, cada uno de ellos con distinta frecuencia central, escalado y orientación.	22
Figura 1.10: Filtros obtenidos mediante el algoritmo POPLS (aprendizaje supervisado).	23
Figura 1.11: Muestras de las diez clases de texturas que elegiremos en este trabajo para su clasificación. (a) Corteza de árbol. (b) Ladrillos. (c) Grietas. (d) Hierba. (e) Grava. (f) Hiedra. (g) Contrachapado. (h) Cielo. (i) Nieve. (j) Agua.....	26
Figura 2.1: La parte real e imaginaria de una senoide compleja; 128x128 píxeles; parámetros: $u_0 = v_0 = 1/80$ ciclos/píxel, $P=0$ grados. Fuente: [14].....	29
Figura 2.2: Curvas de nivel de una envolvente gaussiana; 128x128 píxeles; parámetros: $x_0 = y_0 = 0$. $a=1/50$ píxeles, $b=1/40$ píxeles, $\theta=-45$ grados. Fuente: [14].	30

Figura 2.3: Partes real e imaginaria de una función de Gabor compleja en el dominio espacial; 128x128 píxeles; parámetros: $x_0 = y_0 = 0$, $a = 1/50$ píxeles, $b = 1/40$ píxeles, $\theta = -45^\circ$, $P = 0^\circ$. Fuente: [14].	31
Figura 2.4: La transformada de Fourier del filtro de Gabor; parámetros: $x_0 = y_0 = 0$, $a = 1/50$ píxeles, $b = 1/40$ píxeles, $\theta = -45^\circ$, $F_0 = \sqrt{2} / 80$ ciclos/píxel, $\omega_0 = 45^\circ$, $P = 0^\circ$. Fuente: [14].	32
Figura 2.5: Filtro de Gabor en el dominio frecuencial con sus parámetros. Fuente: [14].	33
Figura 2.6: Espectro de un banco de filtros de Gabor según la Ecuación (2.5) con $\lambda=4$ y resolución de 500x500 píxeles. (a): $\theta = 0$ rad , $s = 1.5$. (b): $\theta = \pi/2$ rad , $s = 1.5$. (c): $\theta = 0$ rad, $s = 2.5$. (d): $\theta = \pi/2$ rad , $s = 2.5$.	35
Figura 2.7: Imagen de muestra (corteza de árbol).	35
Figura 2.8: Imágenes resultantes de procesar la Figura 2.7 con cada uno de los filtros de la Figura 2.6 respectivamente.	36
Figura 2.9: Creación de una base de datos de texturas (un vector de características por cada imagen completa. Fuente: [13].	38
Figura 2.10: Ejemplo de búsqueda de imágenes basada en texturas; resultado para la imagen D5; NTM=12. Fuente: [13].	39
Figura 2.11: Filtro de Gabor tradicional en el dominio del espacio (a) y de la frecuencia (b). Filtro de Gabor circularmente simétrico en el dominio del espacio (c) y de la frecuencia (d). Fuente: [17].	42
Figura 2.12: Representación en el dominio de la frecuencia del conjunto de filtros de Gabor circularmente simétricos, para imágenes de 16x16. Fuente: [17].	43
Figura 2.13: Las 16 texturas empleadas para probar el rendimiento de los filtros de Gabor circularmente simétricos. Fuente: [17].	44
Figura 3.1: Recta que maximiza la varianza de los datos (PCA).	49
Figura 3.2: Conjunto de datos para un problema de clasificación.	58
Figura 3.3: Comparativa de los PCA, PLS y OPLS. Fuente: [27].	59
Figura 4.1: Subconjunto de la base de datos de texturas.	66
Figura 4.2: Contrachapado. (a): resolución original (800x609). (b): resolución final (500x500).	68

Figura 4.3: Tasa de aciertos frente a tamaño de las imágenes.	69
Figura 4.4: Resultado (ampliado) de recortar la textura de Fig. 4.2 a un tamaño de 20x20.....	70
Figura 4.5: (a): Textura ‘hiedra’ en el dominio espacial. (b): Logaritmo de su DFT.	71
Figura 4.6: (a) Espectro de 500x500 puntos. (b) Su promediado a 50x50 puntos.	72
Figura 4.7: Tasa de aciertos en clasificación frente a resolución del promediado de texturas.	73
Figura 4.8: Resultado de promediar el espectro de un filtro de Gabor de 500x500 a 10x10.....	74
Figura 4.9: (a) Textura ‘cielo’. (b) Textura ‘nieve’.	78
Figura 4.10: Esquema de extracción de características de una textura mediante bancos de filtros de Gabor.	79
Figura 4.11: Tasa de aciertos mediante validación cruzada en función del número de filtros de Gabor empleados.....	81
Figura 4.12: Banco de 16 filtros de Gabor (resolución de 10x10).....	82
Figura 4.13: (a): Primer filtro obtenido para la clasificación mediante POPLS. (b): Vista lateral del mismo filtro.	85
Figura 4.14: Texturas de ejemplo. (a): Hierba. (b): Hiedra. (c): Cielo. (d): Nieve.....	87
Figura 4.15: Tasa de aciertos en clasificación frente al número de filtros obtenidos con POPLS.	88
Figura 4.16: 16 primeros filtros obtenidos con POPLS	89
Figura 4.17: Tasa de aciertos de clasificación en validación cruzada frente al número de filtros obtenidos con POPLS y Gabor.....	90
Figura 4.18: $\ \mathbf{Y} - \hat{\mathbf{Y}} \ _F^2$ en el conjunto de entrenamiento frente al número de filtros obtenidos con POPLS y Gabor.	91
Figura 4.19: 16 primeros filtros obtenidos con POPLS (nueva ejecución).	92
Figura A.1: Imagen y su transformada de Fourier. Fuente: [11].	100

Figura A.2: Al multiplicar la imagen de la Figura A.1 por un factor $(-1)^{x+y}$ su espectro resultante queda centrado: las bajas frecuencias se representan en la parte central. Fuente: [11].	101
Figura A.3: Eliminación de la componente continua en una imagen. Fuente: [11].	103
Figura A.4: Función de transferencia de un filtro paso alto ideal.	104
Figura A.5: Resultado de filtrar una imagen por un filtro paso alto ideal. (a): Imagen original. (b): Imagen resultante. Fuente: [35].	105
Figura A.6: Función de transferencia de un filtro paso alto gaussiano.	106
Figura A.7: Resultado de filtrar una imagen por un filtro paso alto gaussiano. (a): Imagen original. (b): Imagen resultante. Fuente: [35].	106
Figura A.8: Función de transferencia de filtros paso bajo. (a): Ideal. (b): Gaussiano.	107
Figura A.9: (a): Imagen original. (b): Resultado de filtrar (a) con un filtro paso bajo ideal. (c): Resultado de filtrar (a) con un filtro paso bajo gaussiano. Fuente: [11].	108
Figura A.10: Función de transferencia de un filtro de banda rechazada	108
Figura A.11: (a): Imagen corrompida por ruido sinusoidal. (b) Resultado de filtrar (a) con un filtro paso banda a la frecuencia de la señal interferente. Fuente: [35].	109

Lista de Tablas

Tabla 2.1: Matriz de confusión para filtros de Gabor circularmente simétricos en texturas rotadas	45
Tabla 3.1: Pseudocódigo de POPLS	57
Tabla 4.1: Posibles fuentes para nuestra base de datos de texturas	64
Tabla 4.2: Ejemplo de matriz de confusión.	76
Tabla 4.3: Matrices de confusión para clasificación de texturas basada en características obtenidas mediante banco de filtros de Gabor. (a): Al no incluir las bajas frecuencias. (b): Al incluirlas.....	77
Tabla 4.4: Matriz de confusión para 16 filtros de Gabor. Tasa De aciertos = 68..93%	83
Tabla 4.5: Matriz de confusión para un solo filtro obtenido con POPLS. Tasa De aciertos = 21.6%.....	86
Tabla 4.6: Matriz de confusión para 16 filtros obtenidos con POPLS. Tasa De aciertos = 68..6%.....	89

Capítulo 1. Introducción a la clasificación de texturas

La clasificación es un proceso por el cual se toman datos para, a partir de sus características, decidir a qué categoría pertenecen. Esta técnica no es sólo propia del campo de la informática, sino un proceso fundamental que se encuentra en casi todas las acciones humanas.

En este capítulo introductorio comenzaremos hablando de la motivación de la clasificación en general, proporcionando algunos ejemplos de sus innumerables aplicaciones, para ir concretando hasta llegar al caso de la clasificación de texturas, que es el tema que nos ocupa.

También analizaremos los bloques básicos de un sistema de reconocimiento de patrones (término más amplio que ‘texturas’ y que emplearemos para referirnos a clases en general). Este sistema consta de:

1. Un sensor, que recoge las observaciones.
2. Algún tipo de preprocesado, que transforme la observación para adecuarla a las fases posteriores.
3. Un sistema de extracción de características, que, de entre todas las variables que caracterizan a la observación, selecciona únicamente las más relevantes (o combinaciones de las mismas).
4. Un clasificador que, basado en las características extraídas, cataloga la medición.

Se hará hincapié en la parte de extracción de características, en la que nos centraremos casi exclusivamente en este proyecto.

Como ya hemos dicho, estamos interesados en un tipo particular de patrones, que son las texturas, y se podrían definir como las características visuales de una superficie. Se consideran texturas los patrones del tipo “hierba”, “nieve” o “corteza de árbol”. Las texturas, al tener propiedades periódicas, facilitan su análisis en el dominio de la frecuencia, lo que permitirá procesarlas mediante bancos de filtros para extraer sus características, que serán componentes frecuenciales, más relevantes. A partir de dichas características se procederá a clasificarlas.

Finalmente señalaremos los objetivos del Proyecto, que a grandes rasgos consisten en comparar dos métodos diferentes de extracción de características: uno fijo y basado en conocimiento experto, y otro mediante aprendizaje máquina supervisado.

1.1. Motivación de la clasificación de patrones

La clasificación es el primer paso que deben dar los sistemas inteligentes frente a los datos que perciben. En el mundo occidental, los cimientos del reconocimiento de patrones se remontan a Platón, más tarde ampliados por Aristóteles, quien hizo distinciones entre la “propiedad esencial” (que es compartida por todos los miembros de una misma clase) y la “propiedad accidental” (que podría diferir entre miembros de la misma clase). El reconocimiento de patrones puede plantearse como el problema de encontrar tal propiedad esencial en una categoría. El estudio de la naturaleza del conocimiento ha sido tema central en la epistemología filosófica [1].

La cantidad de literatura moderna sobre teoría de la decisión y reconocimiento de patrones es, a día de hoy, abrumadora, y se compone de docenas de revistas científicas, miles de libros e innumerables artículos; y continúa creciendo. Algunas disciplinas como la estadística, el aprendizaje máquina y las redes de neuronas establecen los cimientos del reconocimiento de patrones; otras, como la visión computarizada y reconocimiento de habla, se basan en ellos. La psicología perceptual, la ciencia cognitiva, la psicología y la neurociencia analizan cómo se consigue el reconocimiento de patrones en los humanos y otros animales. Esta visión extrema de que todo en la cognición humana, incluido el seguimiento de reglas y la lógica, puede reducirse a reconocimiento de patrones se muestra en [2]. Las técnicas de reconocimiento de patrones se han aplicado a virtualmente todos los campos de la ciencia.

1.2. Aplicaciones de la clasificación de patrones

Hoy en día, las aplicaciones de las técnicas de clasificación son muy numerosas. Vamos a ver los siguientes ejemplos: Reconocimiento del habla, clasificación de textos, reconocimiento de caligrafía y reconocimiento automático de caras. Los dos últimos pertenecen al subcampo del análisis de imágenes, que es hacia lo que este Proyecto está enfocado.

1.2.1. Reconocimiento del habla

El reconocimiento del habla consiste en convertir la voz en texto. Muy relacionado con él está lo que se conoce como “reconocimiento de locutor”, cuando el sistema se entrena para un hablante en particular, como es el caso de la mayoría del software de reconocimiento de ámbito doméstico. Los sistemas de reconocimiento del habla deben estar preparados para identificar a casi cualquier persona, como es el caso de un centro de llamadas diseñado para reconocer

cualquier voz que lo llame, mientras que los sistemas de reconocimiento de locutor reconocen al hablante en particular basándose en su sonido vocal único [3].

Las aplicaciones del reconocimiento del habla abarcan desde el marcado telefónico (teléfonos que reconocen las frases “llamar a casa”, “llamar al trabajo”), aplicaciones domóticas de control basadas en contenido (por ejemplo, buscar un podcast a partir de unas palabras que aparezcan en él), sistema de entrada de datos (como el número de una tarjeta de crédito) o conversores del habla al texto (en editores de texto o redacción de correos electrónicos).

1.2.2. Clasificación de textos

Durante los últimos veinticinco años, el número de documentos de texto en forma digital ha crecido enormemente en tamaño. Como consecuencia, es de vital importancia el ser capaces de organizar y clasificar documentos de forma automática. Existen dos variantes principales de clasificación de texto: agrupamiento de texto y categorización de texto. El primero se ocupa de formar grupos en el conjunto de documentos, mientras que el segundo se puede ver como la tarea de estructurar la base de datos de documentos de acuerdo a grupos que se conocen de antemano [4].

La clasificación de documentos aparece en numerosas aplicaciones, incluyendo el filtrado de correos electrónicos, enrutamiento de correos, filtrado de spam, monitorización de noticias, difusión selectiva de información a clientes, indexado automático de artículos científicos, identificación del género de un documento, supervisión de código, etc. La categorización automática de texto resulta atractiva, ya que organizar los documentos manualmente puede resultar demasiado caro, o inviable, dadas las restricciones temporales de las aplicaciones o el número de documentos disponibles.

1.2.3. Reconocimiento de caligrafía

El reconocimiento de caligrafía es la habilidad de un ordenador para recibir e interpretar caligrafía inteligible como entrada de documentos de papel, fotografías, pantallas táctiles y otros dispositivos. La imagen del texto puede ser detectada de modo “offline”, de un trozo de papel mediante escaneo óptico, o, de manera alternativa, de modo “online”, por ejemplo mediante un sistema con lápiz y superficie electrónicos [5].

El reconocimiento de caligrafía implica el reconocimiento óptico de caracteres. Sin embargo, un sistema de reconocimiento de caligrafía también maneja el formato, realiza segmentación en caracteres y decide entre las palabras más plausibles.

1.2.4. Reconocimiento de caras

Un sistema de reconocimiento facial es una aplicación informática para la identificación o verificación automática a partir de una imagen digital o un fotograma de vídeo. Una de las maneras de hacer esto es mediante la comparación de rasgos faciales seleccionados de la imagen con una base de datos de caras. Se utiliza típicamente en sistemas de seguridad, y puede emplearse en conjunto con otros sistemas biométricos como el reconocimiento de huellas dactilares o el reconocimiento de iris [6].

Los algoritmos de reconocimiento facial identifican las caras mediante la extracción de rasgos de una imagen de la cara del sujeto en cuestión. Por ejemplo, un algoritmo podría analizar la posición relativa, tamaño o forma de los ojos, nariz, pómulos y mandíbula. Estos rasgos luego se emplean en la búsqueda de otras imágenes con los mismos rasgos.

Los algoritmos de reconocimiento se pueden dividir en dos variantes principales: geométricos, que analizan los rasgos distintivos, o fotométricos, que son un

enfoque estadístico que descompone la imagen en valores y los compara con valores de plantillas para eliminar variaciones.

Como parte del análisis facial cabe destacar el análisis de texturas de la piel. Éste hace uso de los detalles visuales cutáneos capturados en imágenes digitales. Con esta técnica se proyectan las líneas, manchas o cualquier otra característica de la piel de la persona a un espacio matemático. Los experimentos muestran que al introducir análisis de texturas de la piel el rendimiento del reconocimiento facial puede mejorar de un 20 a un 25 por ciento [7].

1.3. Un Sistema de clasificación

1.3.1. Generalidades del aprendizaje máquina

Para conseguir clasificar patrones debemos ser capaces de, al observar una muestra x poder decidir a qué clase pertenece. Esto es posible lograrlo por medio de tres métodos alternativos, que son los siguientes:

- Método analítico: Es necesario conocer $P(C_j | x)$, la probabilidad de que, a la vista de una observación x , ésta pertenezca a la clase j para, a partir de ahí, optimizar con respecto a una política de costes.
- Método semianalítico: cuando no se conoce la distribución de $P(C_j | x)$, se procede a estimarla de alguna forma para posteriormente emplear el método analítico.
- Método máquina: no se intenta dar con $P(C_j | x)$. En lugar de ello, se ‘entrena’ el sistema (requiere, por tanto, una base de datos de entrenamiento) para que sea capaz de clasificar patrones de forma automática. El objetivo es obtener una adecuada generalización de las distintas clases.

Considérese una máquina que recibe una secuencia de entradas $x_1, x_2, x_3 \dots$. Estas entradas, que a menudo llamaremos “datos”, podrían corresponder a una imagen en la retina, los píxeles en una cámara o una forma de onda acústica. Podría también corresponder a datos no tan sensoriales, como por ejemplo las palabras en una historia, o la lista de artículos en la cesta de la compra.

Se pueden distinguir dos tipos distintos de aprendizaje máquina:

1. Aprendizaje supervisado, donde la máquina también recibe una lista de salidas deseadas $y_1, y_2, y_3 \dots$, y el objetivo de la máquina es aprender a producir la salida correcta dada una nueva entrada. Es decir, va aprendiendo cómo son los miembros que componen las distintas clases hasta que sea capaz de formar un ‘modelo’ o ‘prototipo’ de los miembros de cada clase.
2. Aprendizaje no supervisado; aquí la máquina recibe las entradas $x_1, x_2, x_3 \dots$, pero no las salidas deseadas. Podría resultar misterioso imaginar qué podría aprender la máquina dado que no recibe retroalimentación de su entorno. Sin embargo, es posible desarrollar un marco formal para el aprendizaje no supervisado basándonos en la noción de que el objetivo de la máquina es construir representaciones de las entradas para que sean utilizadas en la toma de decisiones; de algún modo, el aprendizaje no supervisado puede verse como el encontrar patrones en los datos de entrada e ignorar lo que podría considerarse ruido. Un método de aprendizaje no supervisado es el que se conoce como “agrupamiento” [1].

Para mostrar los bloques constitutivos que componen el aprendizaje máquina emplearemos un ejemplo, que será ficticio, puesto que no se ha llevado a cabo tal estudio, pero será con fines ilustrativos. Consideremos el siguiente problema: la identificación a partir de una fotografía de un árbol para decidir a qué especie pertenece. Como primer enfoque se decide distinguir entre dos tipos de árbol:

roble y encina, empleando detección óptica. Mediante una cámara se toman unas cuantas imágenes de ambas especies (Figura 1.1(a)) y empezamos a percibir diferencias entre ambas, tales como su envergadura, el color y forma de las hojas, el grosor del tronco, etc. Todo ello conforma las características que deberemos explorar para hallar el clasificador. También observamos ruido o variaciones en las imágenes: variaciones en la iluminación, posición del árbol en la fotografía y en definitiva, que no hay dos árboles iguales, incluso perteneciendo a la misma categoría. Dado que hay diferencias objetivas entre encinas y robles, podemos verlo como que ambos tienen distintos modelos: distintos descriptores, que vendrán dados de forma matemática. El objetivo fundamental en la clasificación de patrones es establecer hipótesis sobre la clase a la que pertenecen estos modelos, procesar los datos detectados para eliminar ruido, que no es intrínseco a los modelos, y elegir el modelo en el que mejor encaje cada dato en concreto.

El sistema que lleve a cabo esta tarea específica podría tener la forma mostrada en la Figura 1.1(b). Primero la cámara toma una imagen del árbol. A continuación las señales de la cámara son preprocesadas para simplificar operaciones futuras sin perder información relevante. En particular se podría emplear una operación de segmentación en la que las imágenes de distintos árboles se aislen del fondo. La información de un único árbol se envía entonces a un extractor de características, cuyo propósito sea el de reducir los datos midiendo ciertas “características” o “propiedades”. Estas características (los valores de éstas) son enviadas a un clasificador que evalúa los datos presentados y toma una decisión final en cuanto a la especie a la que pertenece cada instancia. Veamos ahora más en detalle estos bloques constitutivos.

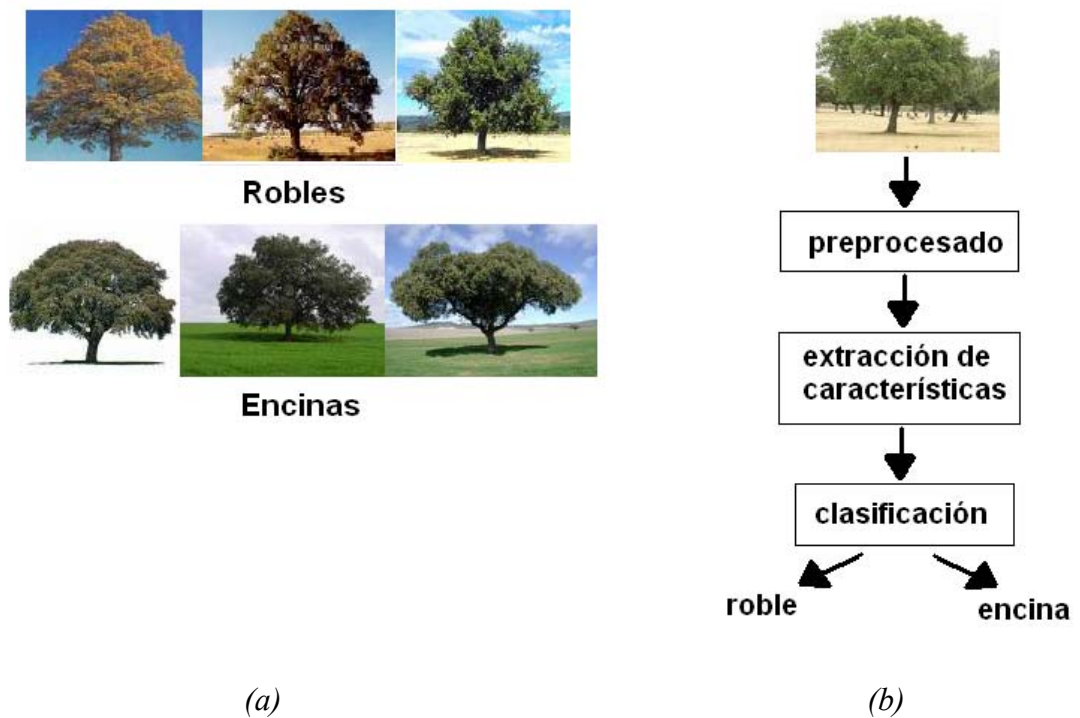


Figura 1.1: Sistema de clasificación de árboles. (a) Un conjunto de árboles de muestra para el entrenamiento. (b) Nueva imagen que deberá ser clasificada.

1.3.2. Preprocesado

El preprocesado es el primer paso que se da en la cadena del sistema de clasificación. Su función es eliminar el ruido, o cualquier otra variación conocida que no sea característica de la clase a la que pertenece, para que éstos no influyan en la clasificación final.

En el ejemplo propuesto, vemos que las imágenes difieren en los terrenos y tonos del cielo de una a otra. Dado que éstos no deberían interferir en la decisión de la especie del árbol se procedería a eliminarlos, obteniendo así todos los árboles sobre, por ejemplo, un fondo blanco. Si además el color del tronco o las hojas no son tan importantes, sino más bien su intensidad, se podrían convertir las imágenes en color a blanco y negro. Otra operación de preprocesado sería

normalizar el tamaño de todos los árboles para que no influya la distancia a la que se ha tomado la foto.

1.3.3. Extracción de características

Tras haber eliminado las variaciones impropias de las clases en los datos de entrada mediante el preprocesado, procedemos a decidir cuáles son las variables en las que fijarnos a la hora de clasificar. De eso se encarga el extractor de características. Una vez obtenidas las características que hemos considerado que serán más decisivas para la clasificación debemos cuantificarlas y mapearlas en un espacio vectorial.

En nuestro ejemplo, para la fase de entrenamiento, podemos por ejemplo observar que las encinas tienen, por lo general, las hojas más oscuras que los robles, por lo que la variable ‘intensidad de hojas’ sería una característica que influiría en la decisión. Otra de ellas podría ser la envergadura del árbol: las encinas parecen ser más anchas que los robles; para que no influya la altura del árbol en la decisión (ya que se pueden haber tomado muestras de árboles de distintas edades), se puede medir dicha envergadura normalizada con respecto a su altura. Si el preprocesado ya se encargó de normalizar las alturas, de modo que fuera la misma para todas las imágenes, no haría falta hacer esto. En las Figuras 1.2, 1.3 y 1.4 se pueden ver representadas, en forma de envolvente de histograma o diagrama de dispersión, dichas variables seleccionadas para las muestras de entrenamiento de la máquina. En la primera de las figuras vemos que no es posible fijar un umbral que separe bien ambas clases sin que se produzca un solape significativo; en la segunda es más fácil separarlas eligiendo como umbral, por ejemplo, el punto de corte entre ambas curvas. En la Figura 1.4 seleccionamos dos características de manera simultánea y, al ser un problema bidimensional, la elección de una frontera de decisión es más compleja que en los casos anteriores.

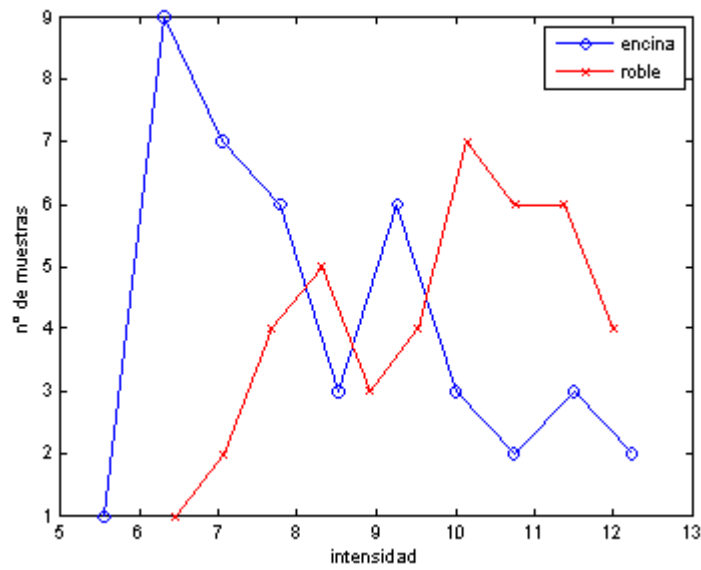


Figura 1.2: Envoltente del histograma que representa la intensidad de las hojas de los árboles con que se entrenará la máquina.

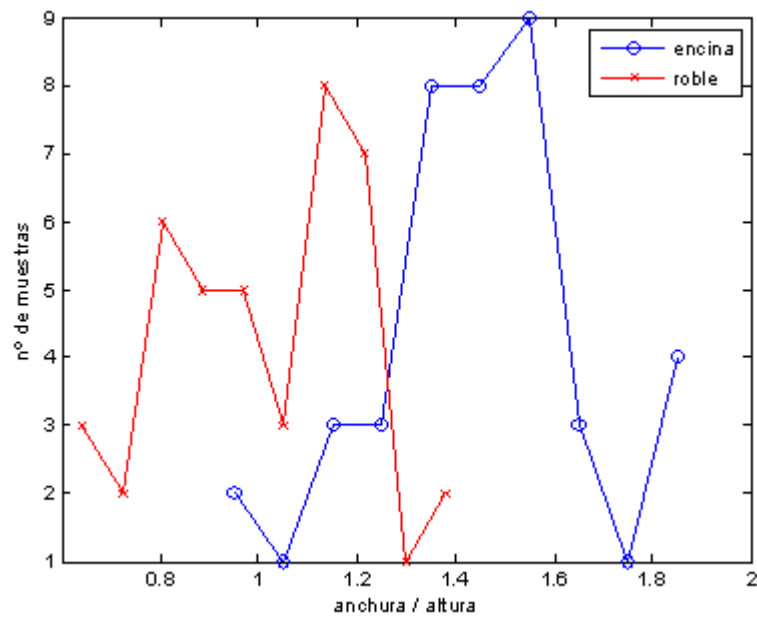


Figura 1.3: Envoltente del histograma que representa la relación ancho / alto de los árboles para cada una de las muestras de entrenamiento.

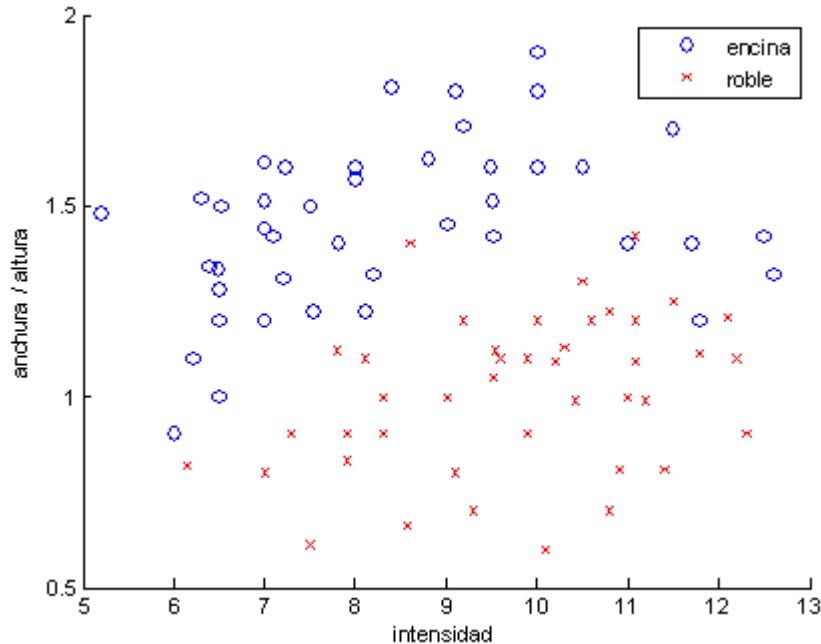


Figura 1.4: Diagrama de dispersión donde se tienen en cuenta simultáneamente tanto la intensidad como la relación ancho / alto de cada árbol para cada clase.

Además de la intensidad de las hojas y su envergadura relativa podríamos incluir otros parámetros como la forma y el número (estimado) de hojas, el grosor del tronco, etc. ¿Cómo sabemos de antemano cuál de estas características funcionará mejor? Algunas características serán redundantes; si por ejemplo la envergadura está perfectamente correlacionada con el grosor del tronco, y con analizar una de las dos características sería suficiente. Pero aún en el caso de que el coste computacional de introducir muchas características no fuera un problema ¿se podrían tener demasiadas características?

En general, si el rendimiento obtenido con un conjunto de características es insuficiente, es normal querer añadir otras nuevas. Pese a que aumentar el número de características aumenta el coste y la complejidad tanto del extractor de características como del clasificador, parece razonable creer que el rendimiento mejorará. Después de todo, lo peor que podría pasar es que el clasificador ignore estas nuevas características, pero si éstas proporcionan información adicional el

rendimiento debe mejorar. Desafortunadamente, se observa con frecuencia en la práctica que, a partir de cierto punto, el introducir características adicionales conlleva un empeoramiento en vez de una mejora de las prestaciones. Esto sucede cuando el número de características es muy elevado en relación con el número de muestras disponibles. Esta paradoja presenta un serio problema para el diseño de clasificadores y se conoce como “explosión dimensional” [1]. La forma de evitar este problema es seleccionar sólo las características más relevantes y dejar fuera las que sean redundantes.

La fase de extracción de características es hacia la que está orientado este proyecto, y se compararán dos métodos diferentes para llevarla a cabo: uno supervisado y el otro, fijo (las características extraídas serán siempre las mismas, sin importar la base de datos); se dispondrá de un conjunto inicial de características (que será muy elevado) para, a partir de ahí, extraer las más relevantes, o, mejor dicho, la mejor combinación lineal de características que maximicen la probabilidad de acierto en la clasificación.

1.3.4. Clasificación

Una vez extraídas las características adecuadas es necesario establecer una frontera de decisión en el conjunto de entrenamiento para determinar qué regiones del espacio vectorial pertenecen a cada clase. La forma de establecer dicha frontera debe tener como objetivo poder predecir la clase a la que pertenece una observación nueva, más que separar perfectamente las muestras de entrenamiento, y, a menudo, la frontera óptima será una solución de compromiso entre simplicidad y prestaciones en el conjunto de entrenamiento.

Partiendo de la Figura 1.4, la forma más sencilla de separar ambas clases sería mediante una recta, llamada frontera de clasificación (Figura 1.5). Las muestras que caigan por encima de dicha recta se clasificarán como encina, y como roble en

caso contrario. Se observa que las clases están separadas de forma bastante aceptable pero en el mundo real una recta no es lo suficientemente precisa como para separar los grupos, por lo que, dependiendo del problema, podrían mejorarse las prestaciones empleando un modelo de frontera más complejo. Si los modelos fueran demasiado complicados el clasificador tendría una frontera de decisión más compleja que una línea recta (Figura 1.6). En este caso las muestras de entrenamiento se separarían perfectamente. No obstante tendríamos un problema con una frontera como la mostrada, pues si bien las muestras de entrenamiento se clasifican sin error ¿qué ocurriría con muestras nuevas? El objetivo era generalizar los modelos y es poco probable que tal umbral de decisión proporcione una buena generalización, puesto que parece que está afinado al conjunto de entrenamiento exclusivamente, en vez de a un modelo verdadero de lo que es un roble y una encina. Este problema se conoce con el nombre de ‘sobreajuste’ [1]. Una solución a esto podría ser obtener más muestras de entrenamiento para hacernos una idea mejor de las verdaderas características subyacentes. Sin embargo, en la mayoría de los problemas de reconocimiento de patrones el número de muestras para entrenamiento que uno puede obtener es limitado. Pero incluso aunque sí que dispusiésemos de un conjunto muy elevado de datos de entrenamiento, si siguiésemos el método de la Figura 1.6 el clasificador produciría un umbral de decisión exageradamente complicado, uno que seguramente no tendría buenas prestaciones para muestras nuevas.

Por tanto, resulta conveniente simplificar la capacidad expresiva del clasificador con la idea de generalizar mejor. Es decir, que no clasifique tan bien en el conjunto de entrenamiento pero sí lo hiciera con muestras futuras, y para ello habría que no tener en cuenta muestras poco comunes de cada clase y ceñirse a las que fueran más representativas. En la Figura 1.7 podemos ver lo que sería una solución de compromiso entre simplicidad y prestaciones en el conjunto de entrenamiento, que seguramente se traduzca en buenas prestaciones para muestras futuras.

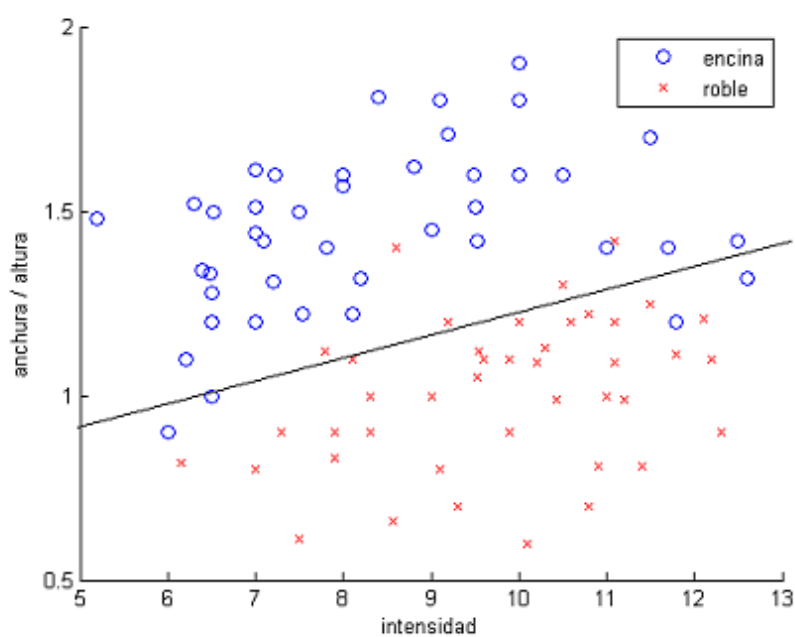


Figura 1.5: Clasificación mediante una frontera lineal entre las dos clases.

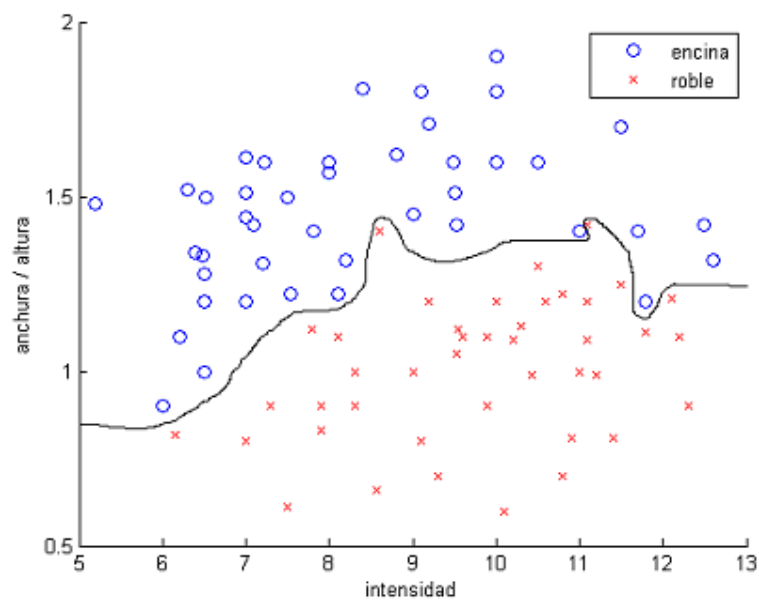


Figura 1.6: Clasificación mediante un modelo de frontera más complicado que el de la Figura 1.5.

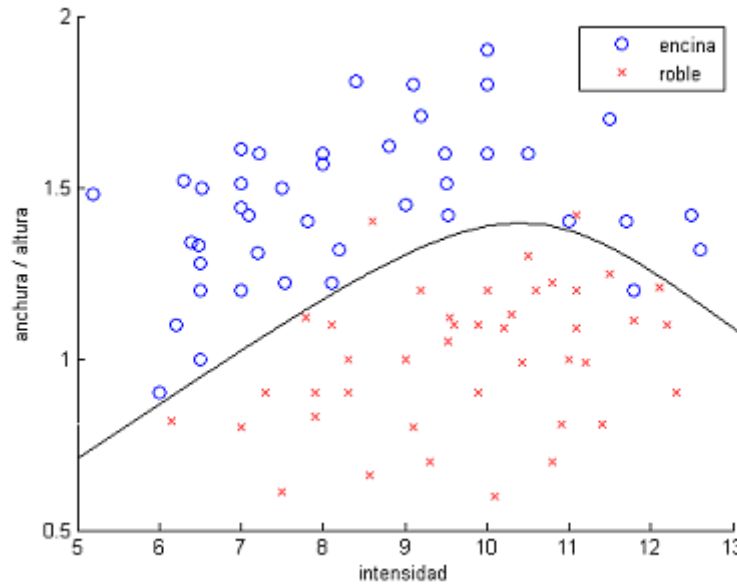


Figura 1.7: Clasificación mediante una frontera de decisión algo más compleja que en la Fig. 1.5 pero no tan compleja como en la Fig. 1.6 con el objeto de obtener una buena generalización.

1.3.5. Conclusiones

En este apartado, y ayudados por el ejemplo expuesto, hemos ilustrado algunos de los planteamientos más importantes a la hora de diseñar un sistema de clasificación máquina:

- La distinción entre los métodos no supervisados y supervisados. En los últimos, además de introducir como información los datos de entrada, se introducen las salidas deseadas. Siempre que el objetivo perseguido sea la estimación o clasificación, el aprendizaje supervisado, tanto en la fase de clasificación como de extracción de características, será el enfoque preferible.
- El ser capaces de extraer las características más relevantes para la clasificación, excluyendo a las redundantes o las menos significativas.

- La elección del tipo de clasificador más apropiado, que deberá tener la complejidad adecuada al tipo de problema. Una elección de clasificador más simple o más compleja de lo necesario incurrirá en peores prestaciones a la hora de clasificar muestras nuevas.

1.4. Clasificación de texturas

La textura se define, según el diccionario de la Real Academia Española, como “estructura, disposición de las partes de un cuerpo, de una obra, etc”. También podemos considerarla como la composición de algo (sobre todo con respecto al tamaño y forma de sus constituyentes más pequeños). Es difícil dar con una definición precisa e inequívoca de “textura” y, sin embargo, todos comprendemos que “grava”, “cuero” o “nieve” constituyen texturas que seríamos capaces, casi siempre, de diferenciar unas de otras.

Las texturas se presentan en todas partes dentro del mundo real; por ello, han sido objeto de estudio durante mucho tiempo dentro del procesamiento de imágenes. Las aplicaciones de los métodos de análisis de texturas se encuentran en campos como la industria, la biomedicina y la detección remota. Además, el reciente alzamiento de todo lo multimedia y la disponibilidad de grandes cantidades de archivos de imagen y vídeo han propiciado que la recuperación de información basada en el contenido se haya convertido en un tema de investigación muy popular [8]. Existen diferentes formas de enfocar el problema del análisis de texturas y, aunque existen relaciones entre ellos, trataremos de clasificar dichos métodos en cuatro grandes grupos como veremos a continuación.

1.4.1. Métodos de clasificación de texturas

1. Métodos estadísticos: estos métodos tienen en cuenta la distribución espacial de los niveles de gris y entre los más conocidos cabe citar a las matrices de co-ocurrencia y la función de autocorrelación [9].
2. Métodos geométricos: éstos consideran la textura compuesta por ‘elementos’ o primitivas. Tras identificar estos elementos la textura puede analizarse bajo dos perspectivas: computar propiedades estadísticas de los elementos extraídos o bien extraer la regla de colocación que describa la textura.
3. Métodos basados en modelos: estos métodos se basan en la construcción de un modelo, cuyos parámetros estimados describirán las cualidades de la textura. Dentro de esta sección destacan los campos aleatorios de Markov y los fractales [10].
4. Métodos de procesamiento de la señal: las técnicas anteriores operan sobre la señal definida en el dominio espacial. Sin embargo, puede resultar más conveniente representar la imagen en otro dominio, de forma que en este nuevo dominio se facilite la extracción de determinadas características. Muchas de estas técnicas consisten en calcular ciertas características de las imágenes tras haber sido filtradas; además, algunas se apoyan en el hecho de que en una primera etapa el cerebro humano lleva a cabo un análisis frecuencial de la imagen. Dentro de este método podemos hacer la siguiente subdivisión:
 - a. Filtros en el dominio espacial: En esta categoría podemos incluir desde métodos que miden la densidad de bordes a través de la magnitud de las respuestas de máscaras de Robert y laplaciana, hasta métodos basados en momentos espaciales y otros obtenidos mediante filtros espaciales y operadores no lineales [11].
 - b. Filtros en el dominio de la frecuencia: Siguiendo los resultados psicovisuales, se han desarrollado sistemas de análisis de texturas de filtrado en el dominio de la frecuencia para obtener distintas características. Cada filtro es selectivo tanto en frecuencia como en

orientación. Estos filtros se analizan en detalle en el Capítulo 2 y el Apéndice A.

1.4.2. Las texturas como patrones periódicos

Las texturas tienen características que las distinguen de otros patrones; una textura tiene características periódicas, es decir, que se repiten en el espacio y eso motiva la realización de la extracción de características en el dominio frecuencial, como vimos en el apartado anterior. Veamos un ejemplo sobre esto.

En la Figura 1.8(a) vemos una textura que clasificaríamos como “ladrillos”. A diferencia de un roble o una encina, esta textura la podríamos replicar indefinidamente horizontal o verticalmente y seguirían siendo “ladrillos”. La Figura 1.8(b) muestra otra textura que se clasificaría como “hierba”. Para poder distinguir entre ladrillos y hierba tenemos que extraer de manera matemática cuáles son las diferencias periódicas entre ambas (recalcamos que aquí solamente analizaremos esas características periódicas y no su color; las texturas se convertirán a blanco y negro como parte de su preprocesamiento). La diferencia fundamental, entonces, entre ladrillos y hierba es que los bloques que se repiten en el espacio son, en el primer caso, más grandes (uno de los ladrillos) que en el segundo (cada brizna de hierba). Entonces podemos concluir, si sólo hubiera que decidir entre ambas clases, que si el periodo fundamental de cada imagen es mayor que cierto umbral (dado en número de píxeles) se trataría de ladrillos, y si fuese menor, sería hierba.

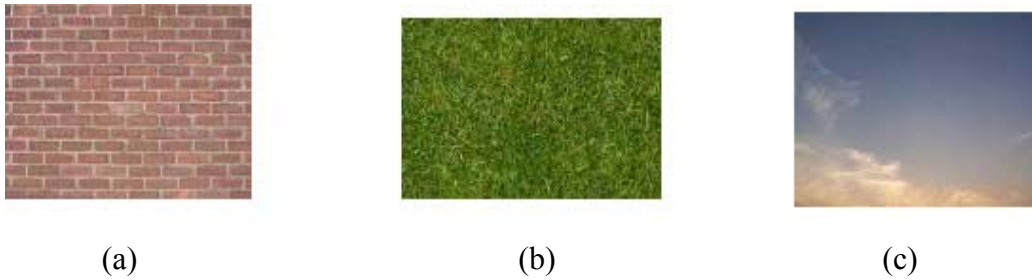


Figura 1.8: Diferentes tipos de texturas: (a) ladrillos, (b) hierba y (c) cielo.

Sin embargo, esta periodicidad que señalamos rara vez se da de manera exacta en la naturaleza: no hay dos ladrillos iguales ni dos briznas de hierba iguales. Es más, en el caso de la Figura 1.7(c) vemos que esta textura, que clasificaríamos como “cielo”, es claramente no periódica. Sin embargo, podríamos considerarla aproximadamente periódica si la tomamos como una textura de periodo fundamental grande (una nube de tamaño medio rodeada de cielo azul). Esta sería una suposición poco precisa, pero aún así, en su análisis espectral veremos que se obtendrán resultados aceptables. Además, parte del objeto del estudio es elegir texturas que se den en la realidad y que sean clasificables por personas, no sólo mediante máquinas.

1.4.3. Aplicaciones de la clasificación de texturas

A continuación comentamos tres ejemplos donde la clasificación de texturas encuentra su aplicación. Si bien éstos podrían haberse incluido en la Sección 1.2 (ya que las texturas son, al fin y al cabo, un tipo particular de patrones) optamos por incluirlos aquí para concretar ya más las aplicaciones que el objeto de estudio de este Proyecto podría tener.

a) Aplicaciones biomédicas: detección de células enfermas

Dado que las células enfermas (en, por ejemplo, tumores) poseen una textura diferente de las normales, para facilitar su detección en fases más tempranas de las que un ser humano podría ser capaz a simple vista, sería de gran ayuda disponer

de un sistema de detección automático que las localizara y así poder tratar la enfermedad a tiempo.

b) Aplicaciones industriales: detección de fallos en materiales textiles

Un método de supervisión automático en una fábrica textil para detectar fallos o roturas en sus tejidos también sería de utilidad [12]. Un tejido que no cumpla con sus cualidades de periodicidad en el momento de la inspección podría hacer saltar una alarma para descartarlo antes de que llegue al usuario final.

c) Aplicaciones en teledetección: cartografía

Mediante una fotografía aérea se puede extraer información acerca del terreno observado. Por ejemplo, si quisiéramos localizar los aparcamientos al aire libre que hay en una ciudad [13] éstos se podrían caracterizar en forma de textura (compuesta por la disposición de coches aparcados en batería unos junto a otros). De esta manera podría filtrarse de manera rápida todo lo que no fuera ‘aparcamientos de coches’, ya que efectuar esta tarea de forma visual sería muy difícil dada la complejidad de estructuras que componen la foto aérea de una ciudad.

1.4.4. Extracción de características mediante filtros de Gabor

Hasta ahora la forma más común de analizar las propiedades periódicas de las texturas se lleva a cabo procesándolas por un banco de filtros especiales denominados ‘filtros de Gabor’ [14]. Estos son, a grandes rasgos, filtros paso banda, que dan información de las frecuencias espaciales que contienen las imágenes, así como de su orientación.

En la Figura 1.9 se muestra el espectro de dos filtros de Gabor. Cada uno de ellos viene definido por tres parámetros:

- La frecuencia central, que resaltará si la textura tiene componentes apreciables a esa frecuencia espacial.
- El escalado, que dejará pasar más o menos frecuencias en torno a la frecuencia central.
- La orientación; ya que en el dominio de las imágenes (bidimensionales) las frecuencias vienen definidas como vectores, con módulo y fase. Esta fase u orientación distinguirá, por ejemplo las rayas horizontales de las verticales, a igual frecuencia.

Se analizará con mayor detalle todo lo relativo al tratamiento digital de imágenes en el Capítulo 2, incluido el diseño y elección de los bancos de filtros de Gabor.

Los filtros de Gabor se elegirán de antemano, antes de analizar nuestra base de datos de texturas para su posterior clasificación; se trata, como ya hemos visto, de un método de aprendizaje no supervisado. ¿Pero sería posible elegir los filtros teniendo en cuenta dicha base de datos para así aumentar la probabilidad de acierto al decidir la clase a la que pertenecen (y efectuar, por tanto, un aprendizaje supervisado)? A continuación propondremos un método para lograrlo.

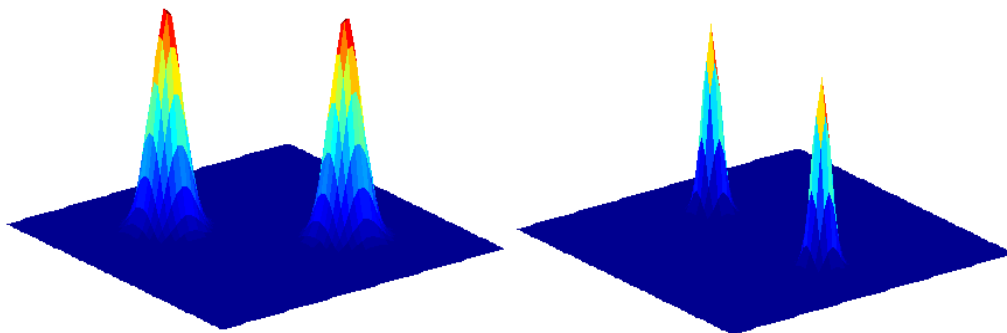


Figura 1.9: Dos filtros de Gabor, cada uno de ellos con distinta frecuencia central, escalado y orientación.

1.4.5. Extracción de características mediante POPLS

El método propuesto de extracción de características supervisadas, y que se explicará en detalle en el tercer capítulo, se denomina POPLS (Positive-constrained Orthonormalized Partial Least Squares) [15], en el cual la forma de obtener los filtros corresponde a un problema de optimización. No entraremos ahora en los detalles del algoritmo, pero sí adelantamos la forma de los filtros que pueden resultar de su implementación (Figura 1.10). Estos dos filtros mostrados indican que, tras analizar todas las imágenes de entrenamiento, las componentes más importantes a la hora de clasificar una imagen nueva son las resaltadas en la Figura 1.10(a), y, si quisiéramos mejorar aún más las prestaciones, elegiríamos además las componentes resaltadas en la Figura 1.10(b).

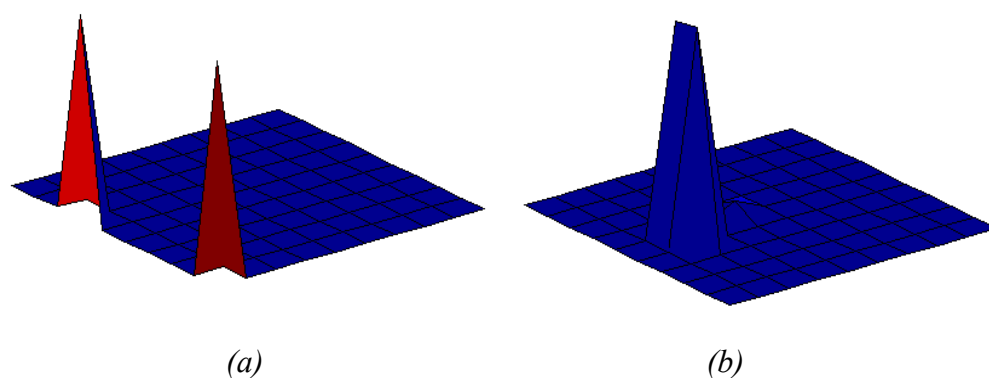


Figura 1.10: Filtros obtenidos mediante el algoritmo POPLS (aprendizaje supervisado).

Este método de extracción de características, a la hora de clasificar, tendrá unas prestaciones distintas a las de simplemente escoger unos filtros predeterminados, y será nuestro propósito averiguar si dichas prestaciones superan a las proporcionadas por el banco de de filtros fijos de Gabor.

1.5. Objetivo del Proyecto

Lo que se pretende en este trabajo es estudiar distintos métodos de extracción de características en problemas de clasificación de texturas. Los pasos que debemos dar para conseguirlo son los siguientes:

- 1) En primer lugar hay que seleccionar una base de datos de texturas sobre la que trabajar. Las clases que finalmente se fijaron fueron las diez siguientes (Figura 1.11):
 - a) Corteza de árbol
 - b) Ladrillos
 - c) Grietas
 - d) Hierba
 - e) Grava
 - f) Hiedra
 - g) Contrachapado
 - h) Cielo
 - i) Nieve
 - j) Agua
- 2) Sobre las imágenes habrá que efectuar un preprocesado que consistirá, a grandes rasgos, en lo siguiente:
 - a) Recortarlas para que todas tengan el mismo tamaño
 - b) Convertirlas a blanco y negro (el color no debe ser una característica de selección pero sí su intensidad).
 - c) Efectuar su transformada de Fourier para trabajar con la textura en el dominio de la frecuencia.
- 3) Una vez hecho esto se seleccionarán las características mediante la información resultante de procesar las imágenes con un banco de filtros. Dichos filtros pueden ser de dos tipos:

- Fijos, de tipo Gabor.
- Obtenidos por aprendizaje supervisado, mediante POPLS.

Compararemos ambos métodos y analizaremos si merece la pena el empleo del entrenamiento supervisado, que es computacionalmente más costoso, a la hora de clasificar texturas, o si, por el contrario, los bancos de filtros de Gabor predefinidos tienen mejores prestaciones, al estar relacionados con el sistema óptico humano de detección.

- 4) Finalmente, se construirá el clasificador más sencillo posible, mediante regresión lineal múltiple (del estilo de la Figura 1.5), para decidir la clase a la que pertenece cada imagen. La elección del clasificador más sencillo responde a que nuestro objeto de estudio no es éste, sino la extracción de características.

Por todo ello, el resto del Proyecto fin de Carrera se estructura de la siguiente manera:

- En el Capítulo 2 se hablará de todo lo referente a los filtros de Gabor y a su uso en la clasificación de texturas.
- En el Capítulo 3 hablaremos de métodos supervisados de extracción de características, en particular de POPLS y explicaremos por qué seleccionaremos este método para la extracción de características de texturas.
- El Capítulo 4 veremos toda la fase de experimentos. Empezamos explicando cómo elegir nuestra base de datos, qué preprocesado dar a las texturas, etc. para luego probar las prestaciones que tiene el extraer características mediante filtros de Gabor; seguidamente haremos lo mismo pero de manera supervisada con POPLS y, finalmente, compararemos los resultados de ambos métodos.
- El Capítulo 5 está dedicado a las conclusiones y líneas futuras.

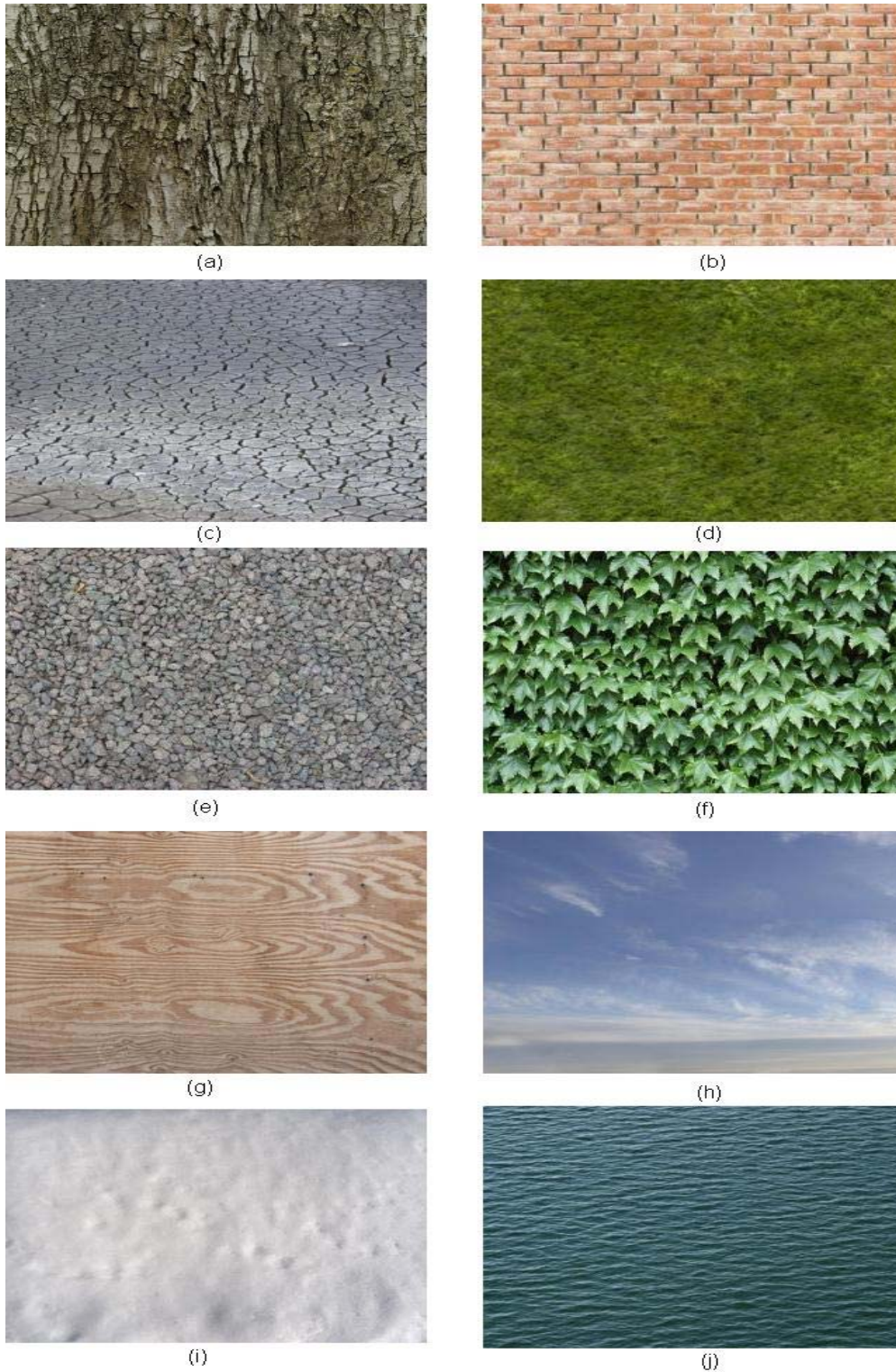


Figura 1.11: Muestras de las diez clases de texturas que elegiremos en este trabajo para su clasificación. (a) Corteza de árbol. (b) Ladrillos. (c) Grietas. (d) Hierba. (e) Grava. (f) Hiedra. (g) Contrachapado. (h) Cielo. (i) Nieve. (j) Agua.

Capítulo 2. Extracción de características mediante filtros de Gabor

En el capítulo anterior hemos explicado de forma general la estructura de un sistema de clasificación. De entre todas las fases que lo componen (preprocesamiento, extracción de características y clasificación) la que más nos interesa en este Proyecto es la de extracción de características. En este capítulo veremos cómo extraer características de texturas, empleando bancos de filtros de Gabor.

Los filtros de Gabor son de uso común en el análisis de texturas. Existen fuertes indicios de que las células simples del córtex visual se pueden modelar mediante filtros de Gabor [16], sintonizados para detectar diferentes orientaciones y escalas. Estos filtros, cuando se aplican a imágenes, producen características que son la base para muchas definiciones de la textura.

Comenzaremos describiendo de forma matemática los filtros de Gabor, pero para poder comprender su desarrollo serán necesarios los conceptos básicos sobre el tratamiento digital de imágenes, que se encuentran en el Apéndice A. En la segunda parte del capítulo veremos ejemplos prácticos de análisis de texturas mediante filtros de Gabor.

2.1. Fundamentos de los filtros de Gabor bidimensionales

La función de Gabor compleja no es más que el producto de una señal sinusoidal y una señal gaussiana, y viene dada por

$$g(x, y) = s(x, y)w_r(x, y) \quad (2.1)$$

donde $s(x, y)$ es la senoide compleja, denominada ‘portadora’, y $w_r(x, y)$ es la gaussiana, también llamada ‘envolvente’. A continuación detallamos cada componente por separado, así como algunas de las propiedades de esta función. Lo haremos para el caso bidimensional, ya que lo que nos interesa en este proyecto son las imágenes. Si se desea ver cómo son los filtros de Gabor en una dimensión se puede consultar el Apéndice B, ya que muchos de los conceptos que revisemos a continuación son extensiones del caso unidimensional.

2.1.1. La portadora sinusoidal compleja

El primer factor que compone la función de Gabor, según (2.1), se define como

$$s(x, y) = \exp(j(2\pi(u_0x + v_0y) + P))$$

donde (u_0, v_0) y P son la frecuencia espacial y la fase de la senoide respectivamente.

Podemos pensar en esta senoide como dos funciones reales separadas, convenientemente situadas en la parte real e imaginaria de una función compleja, de la siguiente forma:

$$\text{Re}(s(x, y)) = \cos(2\pi(u_0x + v_0y) + P)$$

$$\text{Im}(s(x, y)) = \text{sen}(2\pi(u_0x + v_0y) + P)$$

Los parámetros u_0 y v_0 definen la frecuencia espacial de la senoide en coordenadas cartesianas. La frecuencia espacial también puede ser expresada en coordenadas polares con magnitud F_0 y dirección ω_0 :

$$F_0 = \sqrt{u_0^2 + v_0^2}$$

$$\omega_0 = \tan^{-1} \left[\frac{v_0}{u_0} \right]$$

o de manera inversa:

$$u_0 = F_0 \cos \omega_0$$

$$v_0 = F_0 \text{sen} \omega_0$$

Usando esta última representación, la senoide compleja se puede escribir como

$$s(x, y) = \exp(j(2\pi F_0(x \cos \omega_0 + y \text{sen} \omega_0) + P))$$

y se puede ver representada en la Figura 2.1.

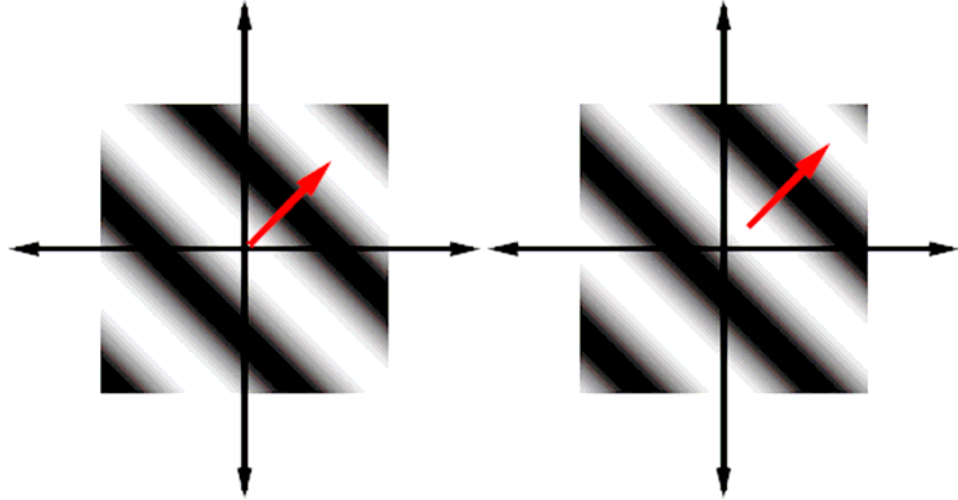


Figura 2.1: La parte real e imaginaria de una senoide compleja; 128x128 pixeles; parámetros: $u_0 = v_0 = 1/80$ ciclos/pixel, $P=0$ grados. Fuente: [14]

2.1.2. La envolvente gaussiana

En la Figura 2.2 podemos ver representada la envolvente gaussiana; la expresión matemática que caracteriza dicha función es:

$$w_r(x, y) = K \exp(-\pi(a^2(x-x_0)_r^2 + b^2(y-y_0)_r^2))$$

Donde (x_0, y_0) son las coordenadas del máximo de la función, a y b son parámetros de escala de la Gaussiana, K es un factor de escala y el subíndice $_r$ denota una operación de rotación tal que

$$(x-x_0)_r = (x-x_0)\cos\theta + (y-y_0)\sin\theta$$

$$(y-y_0)_r = -(x-x_0)\sin\theta + (y-y_0)\cos\theta$$

donde θ es el ángulo de la rotación, que también es un parámetro que caracteriza a la función.

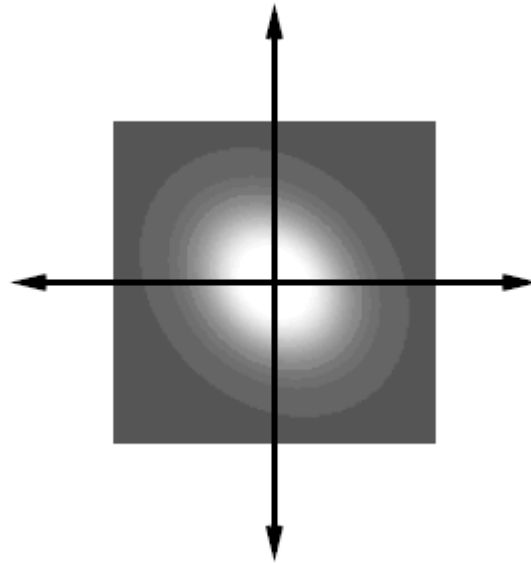


Figura 2.2: Curvas de nivel de una envolvente gaussiana; 128x128 píxeles; parámetros: $x_0 = y_0 = 0$. $a=1/50$ píxeles, $b=1/40$ píxeles, $\theta=-45$ grados. Fuente: [14].

2.1.3. La función de Gabor compleja

En base a lo revisado a los dos apartados previos, la función compleja de Gabor se forma combinando la portadora sinusoidal compleja con la envolvente gaussiana. La función de Gabor compleja se define por los siguientes 9 parámetros:

- K : Escalado de la magnitud de la envolvente gaussiana.
- (a, b) : Escalado de los dos ejes de la envolvente gaussiana.
- θ : Ángulo de rotación de la envolvente gaussiana.
- (x_0, y_0) : Posición del pico de la envolvente gaussiana
- (u_0, v_0) : Frecuencias espaciales de la portadora sinusoidal en coordenadas cartesianas. También se puede expresar en coordenadas polares como (F_0, ω_0) .
- P : Fase de la portadora sinusoidal.

Cada función de Gabor consta de dos funciones en cuadratura (desfasadas 90 grados), situadas en la parte real e imaginaria de una función compleja, como se ve en la Figura 2.3.

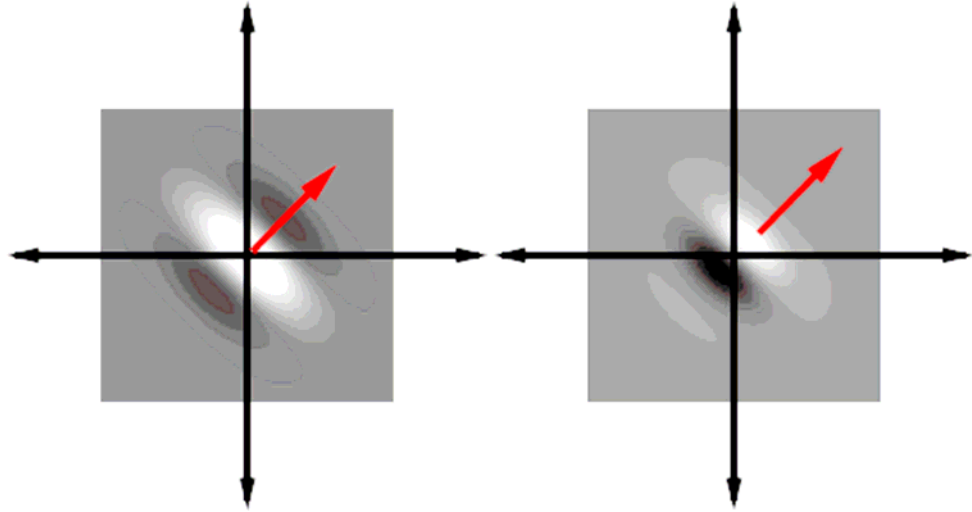


Figura 2.3: Partes real e imaginaria de una función de Gabor compleja en el dominio espacial; 128x128 píxeles; parámetros: $x_0 = y_0 = 0$, $a = 1/50$ píxeles, $b = 1/40$ píxeles, $\theta = -45^\circ$, $P = 0^\circ$. Fuente: [14].

La expresión de la función de Gabor en el dominio espacial es

$$g(x, y) = K \exp(-\pi(a^2(x - x_0)_r^2 + b^2(y - y_0)_r^2)) \exp(j(2\pi(u_0x + v_0y) + P)) \quad (2.2)$$

o, en coordenadas polares,

$$g(x, y) = K \exp(-\pi(a^2(x - x_0)_r^2 + b^2(y - y_0)_r^2)) \exp(j(2\pi F_0(x \cdot \cos \omega_0 + y \cdot \sin \omega_0) + P))$$

La transformada de Fourier de este filtro de Gabor se puede visualizar en la Figura 2.4 (El máximo está en la frecuencia espacial de la senoide compleja) y se expresa como

$$\hat{g}(u, v) = \frac{K}{ab} \exp(j(-2\pi(x_0(u - u_0) + y_0(v - v_0)) + P)) \exp\left(-\pi\left(\frac{(u - u_0)_r^2}{a^2} + \frac{(v - v_0)_r^2}{b^2}\right)\right)$$

o, en coordenadas polares,

$$\text{Magnitud: } |\hat{g}(u, v)| = \frac{K}{ab} \exp\left(-\pi\left(\frac{(u-u_0)_r^2}{a^2} + \frac{(v-v_0)_r^2}{b^2}\right)\right) \quad (2.3)$$

$$\text{Fase: } \angle \hat{g}(u, v) = -2\pi(x_0(u-u_0) + y_0(v-v_0)) + P$$

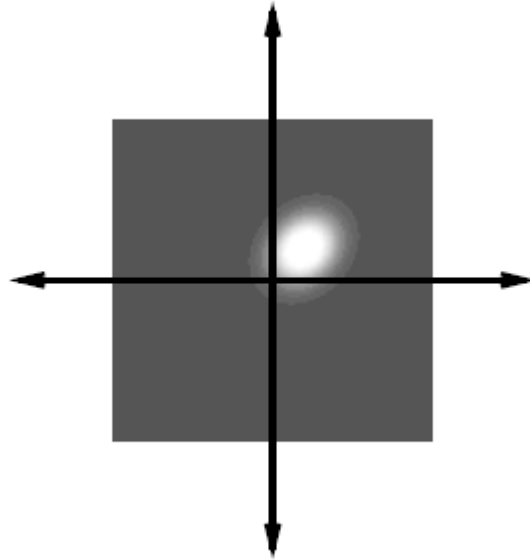


Figura 2.4: La transformada de Fourier del filtro de Gabor; parámetros: $x_0 = y_0 = 0$, $a = 1/50$ píxeles, $b = 1/40$ píxeles, $\theta = -45^\circ$, $F_0 = \sqrt{2} / 80$ ciclos/píxel, $\omega_0 = 45^\circ$, $P = 0^\circ$. Fuente: [14].

2.1.4. Perfil de magnitud mitad

Una forma práctica de representar la función compleja de Gabor es recalando la elipse que forma el perfil de magnitud mitad del lóbulo principal. El conjunto de puntos, en el dominio de la frecuencia, con magnitud igual a un medio de la magnitud de pico se obtiene como vamos a ver. Puesto que el valor de pico se tiene para $(u, v) = (u_0, v_0)$, y la magnitud de pico es igual, según la ecuación (2.3),

a $\frac{K}{ab}$, tan sólo necesitamos encontrar el conjunto de puntos (u, v) con magnitud

$\frac{K}{2ab}$, y eso se consigue como

$$\frac{K}{2ab} = \frac{K}{ab} \exp\left(-\pi\left(\frac{(u-u_0)_r^2}{a^2} + \frac{(v-v_0)_r^2}{b^2}\right)\right) - \log 2 = -\pi\left(\frac{(u-u_0)_r^2}{a^2} + \frac{(v-v_0)_r^2}{b^2}\right)$$

o, equivalentemente:

$$\left(\frac{(u-u_0)_r}{aC} \right)^2 + \left(\frac{(v-v_0)_r}{bC} \right)^2 = 1 \quad (2.4)$$

donde $C = \sqrt{\frac{\log 2}{\pi}} = 0.46971864 \approx 0.5$

La ecuación (2.4) es una elipse centrada en (u_0, v_0) rotada un ángulo θ con respecto al eje u . Los ejes principal y secundario tienen una longitud de $2aC \approx a$ y $2bC \approx b$ respectivamente.

En la Figura 2.5 vemos el perfil de magnitud mitad del filtro de Gabor en el dominio de la frecuencia y sus parámetros: a es eje principal y b , el secundario.

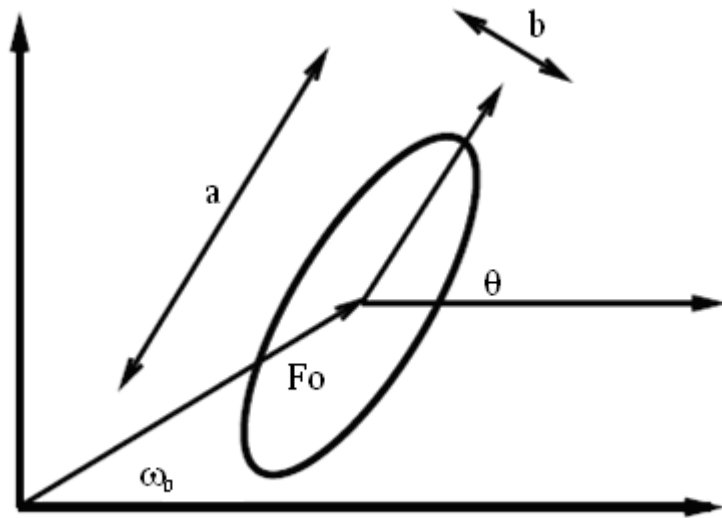


Figura 2.5: Filtro de Gabor en el dominio frecuencial con sus parámetros. Fuente: [14].

2.1.5. Filtro de Gabor simplificado y ejemplos

Veamos ahora en la práctica cómo se transforma una textura al pasar por un banco de filtros de Gabor. Para que sea un ejemplo sencillo primero consideraremos como base de nuestros filtros

$$g(x, y) = \exp\left(-\frac{1}{2}\left(\frac{x_r^2 + y_r^2}{s^2}\right)\right) \cos\left(\frac{2\pi}{\lambda} x_r\right) \quad (2.5)$$

$$x_r = x \cos \theta + y \sin \theta$$

$$y_r = -x \sin \theta + y \cos \theta$$

que es una versión simplificada de (2.2). Además de ser una función real:

- $K=1$: El escalado se mantendrá constante
- $(x_0, y_0) = (0, 0)$ La posición espacial del máximo permanecerá en el origen
- $a = b = \frac{1}{s}$ La envolvente gaussiana no tendrá excentricidad
- $P=0$ La portadora sinusoidal no tendrá desfase

Asimismo, la portadora recorre el sentido de x_r exclusivamente. Este modelo simplificado de filtro de Gabor será el que usemos en todos nuestros experimentos a lo largo del Capítulo 4.

Si consideramos únicamente la variación de dos parámetros, θ y s , y damos dos valores a cada uno de ellos obtenemos cuatro filtros que, en el dominio de la frecuencia, serán reales y positivos y se muestran en la Figura 2.6. Nótese en dichos filtros que un valor de ensanchado s más elevado produce filtros más selectivos (estrechos) en frecuencia. Dichos filtros, al aplicarse a una textura cualquiera (Fig. 2.7), producen las imágenes mostradas en las Figuras 2.8(a-d). El filtrado de imágenes se explica en la Sección A.2.

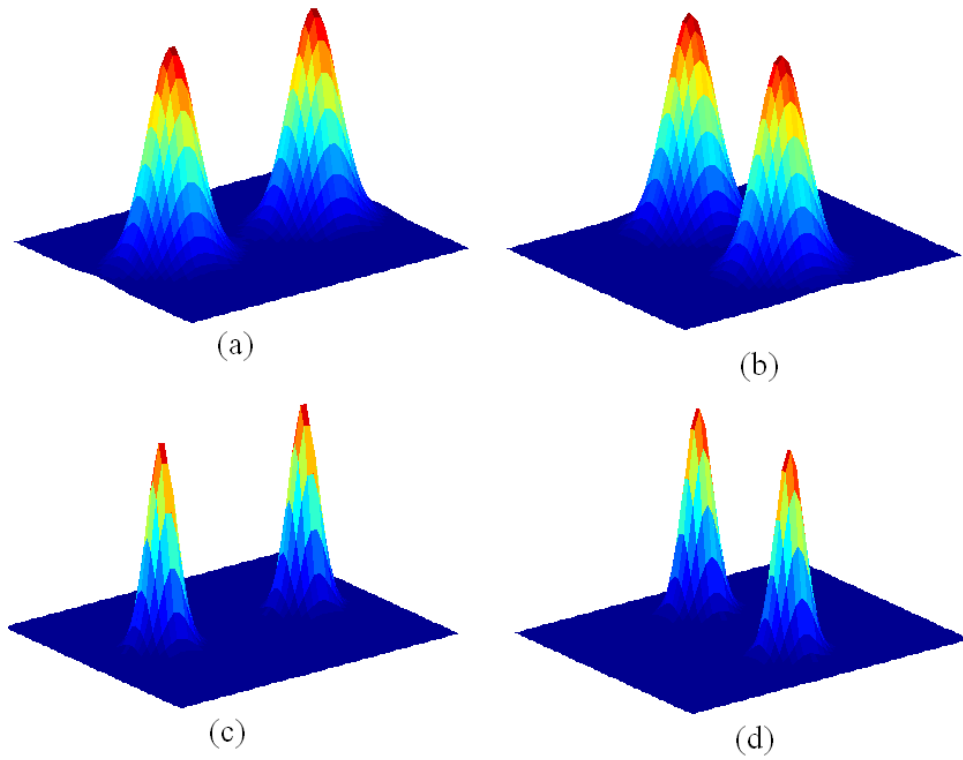


Figura 2.6: Espectro de un banco de filtros de Gabor según la Ecuación (2.5) con $\lambda=4$ y resolución de 500x500 píxeles. (a): $\theta = 0$ rad, $s = 1.5$. (b): $\theta = \pi/2$ rad, $s = 1.5$. (c): $\theta = 0$ rad, $s = 2.5$. (d): $\theta = \pi/2$ rad, $s = 2.5$.



Figura 2.7: Imagen de muestra (corteza de árbol).

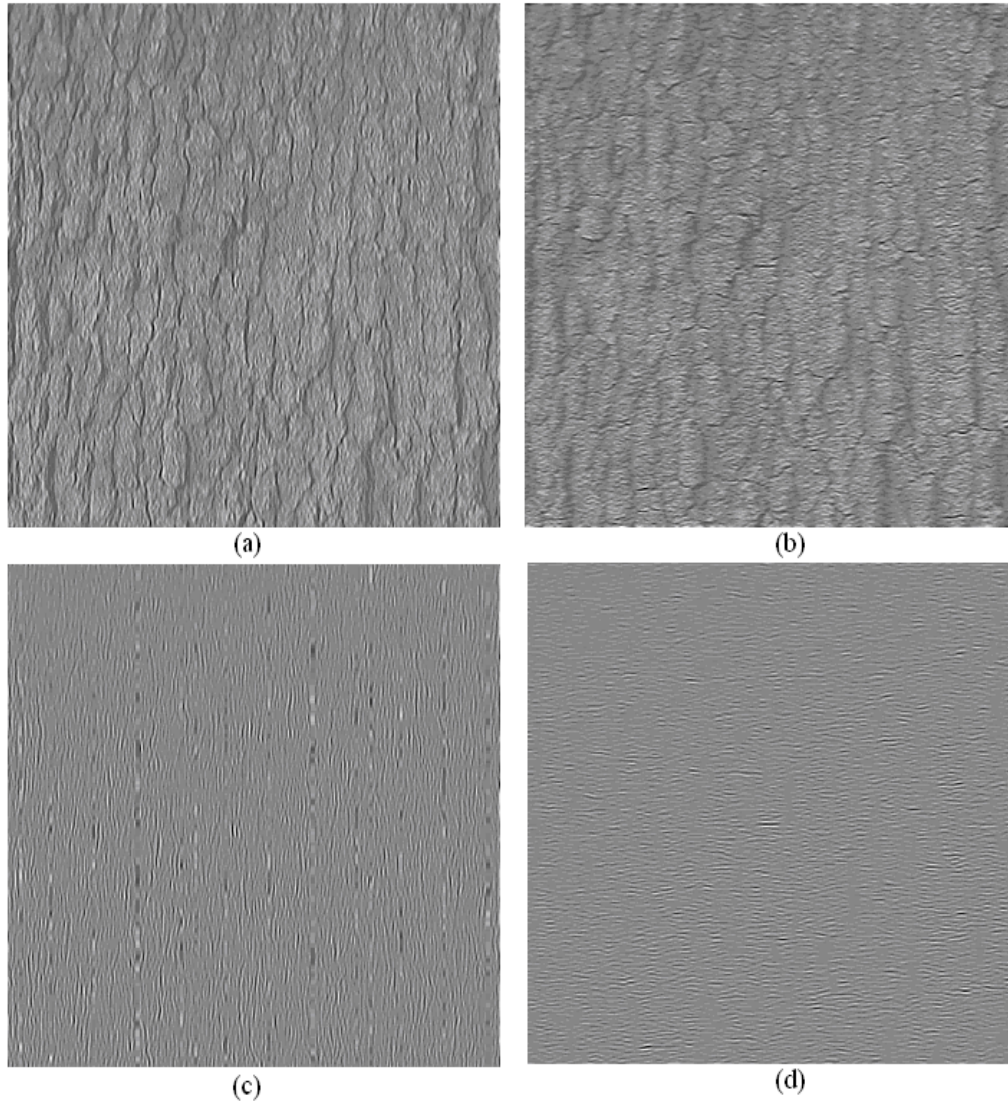


Figura 2.8: Imágenes resultantes de procesar la Figura 2.7 con cada uno de los filtros de la Figura 2.6 respectivamente.

Analizando en detalle las imágenes de la Figura 2.8 observamos lo siguiente:

- Mediante la orientación del filtro (θ), se seleccionan las componentes frecuenciales en una determinada dirección de la textura, que en este ejemplo son horizontales (Fig. 2.8(b),(d)) o verticales (Fig. 2.8(a),(c)).

- Mediante el escalado del filtro (s) se selecciona el rango de frecuencias cercanas a la central que componen la textura. Esta selección es más fina en las Figs. 2.8(c),(d) que en la Fig. 2.8(a),(b).

2.2. Uso de filtros de Gabor en la clasificación de texturas

Hemos visto cómo son de forma matemática los filtros de Gabor y estudiado sus propiedades. Conviene ahora ver la utilidad de estos filtros en la práctica. Se analizarán un par de estudios realizados en [17] y [13] donde se emplean filtros de Gabor para la clasificación de texturas. El primero consiste en cuantificar el parecido entre texturas; el segundo, en clasificar texturas independientemente de la orientación con la que se presenten.

2.2.1. Búsqueda de similitud de texturas

En el siguiente ejemplo, llevado a cabo en [13], se pretende diseñar un sistema que, dada una textura, halle las que más se le parecen, estableciendo un número determinado de mejores aciertos (NTM, Number of Top Matches) definidos por el usuario; se emplean cuatro valores de escalado (s) y seis orientaciones (θ), con lo que tenemos 24 filtros distintos que descomponen la imagen en 24 imágenes filtradas. Por cada imagen filtrada se extraen dos características: la media y la desviación típica. Por tanto, tras la descomposición de Gabor, se extraen 48 características por cada imagen de entrada.

Partimos de 112 texturas. Como cada imagen se puede representar por un vector de 1x48 características, nuestra base de datos será una matriz de tamaño 112x48. Este proceso se ilustra en la Figura 2.9.

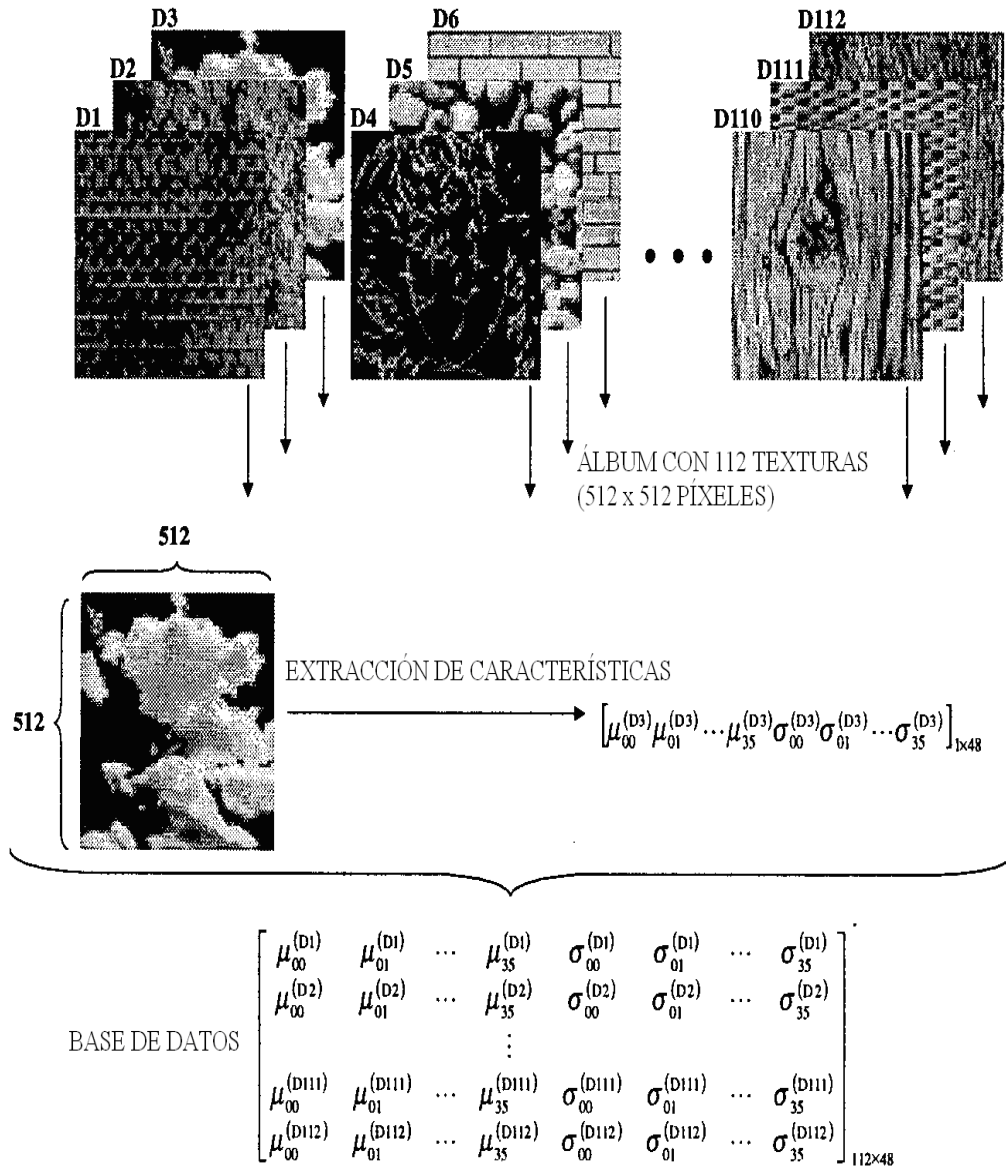


Figura 2.9: Creación de una base de datos de texturas (un vector de características por cada imagen completa. Fuente: [13].

Después de crear la base de datos y extraer las características podemos calcular la medida de similitud de una imagen con las demás. La medida de similitud, (sm), se calcula de acuerdo a la siguiente fórmula para la distancia euclídea normalizada:

$$sm(i, j) = \sum_{s=1}^{s4} \sum_{\theta=01}^{\theta6} \left| \frac{\mu_{s\theta}^{(i)} - \mu_{s\theta}^{(j)}}{d(\mu_{s\theta})} \right| + \left| \frac{\sigma_{s\theta}^{(i)} - \sigma_{s\theta}^{(j)}}{d(\sigma_{s\theta})} \right|$$

donde $\mu_{s\theta}^{(i)}$ y $\sigma_{s\theta}^{(i)}$ son, respectivamente, el valor medio y la desviación típica de la imagen i con escalado s y orientación θ ; $d(\mu_{s\theta})$ es la desviación típica de los valores de la media en toda la base de datos, y $d(\sigma_{s\theta})$ es la desviación típica de las desviaciones típicas de toda la base de datos. Con este enfoque obtenemos 112 medidas de similitud distintas para cada imagen concreta. Las medidas de similitud son entonces clasificadas en orden ascendente. El NTM determina cuántos resultados para cada imagen solicitada obtendremos.

Consideremos por ejemplo un NTM=12 (Fig. 2.10) La imagen de la izquierda es la solicitada, y el resto son las imágenes obtenidas en orden de similitud (de izquierda a derecha y de arriba a abajo).

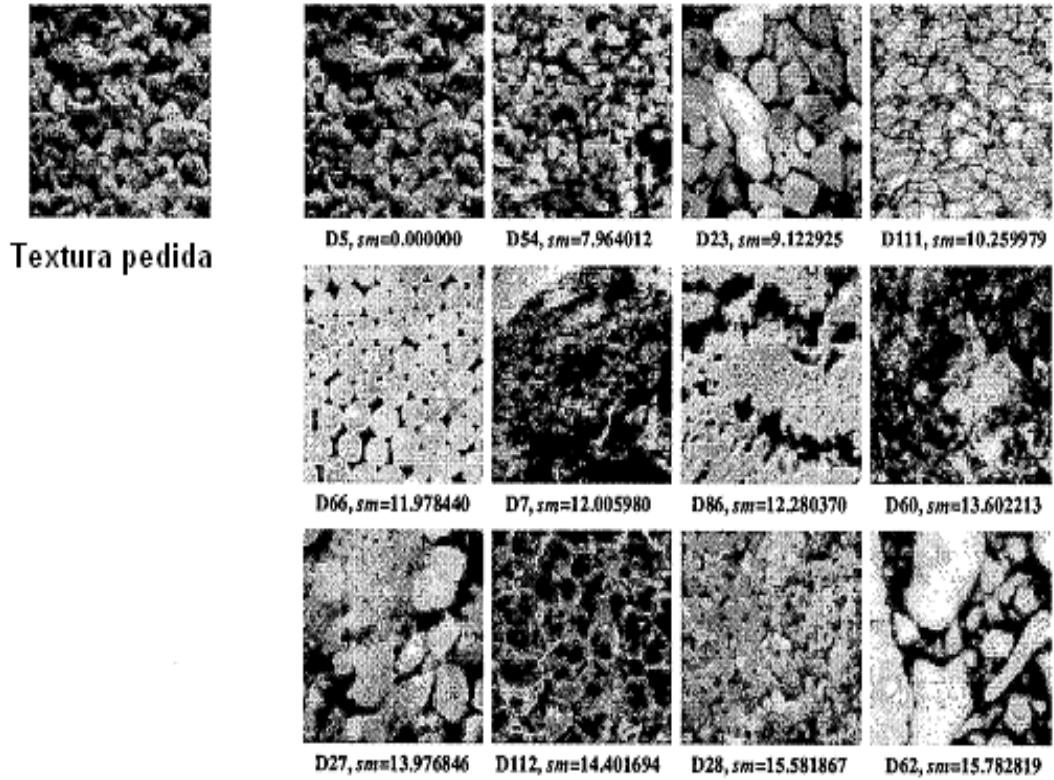


Figura 2.10: Ejemplo de búsqueda de imágenes basada en texturas; resultado para la imagen D5; NTM=12. Fuente: [13].

Es interesante ver que el mejor resultado (el más similar) se corresponde exactamente con la imagen solicitada. Este hecho verifica la correcta implementación: si la imagen solicitada está incluida en la base de datos debe obtenerse como el mejor de los resultados (su medida de similitud debe ser igual a cero), ya que las características de la textura solicitada son idénticas a las características de la misma imagen de la base de datos.

2.2.2. Clasificación de imágenes invariantes a la rotación

En este experimento, llevado a cabo por [17], se pretende usar un nuevo conjunto de filtros de Gabor circularmente simétricos para conseguir insensibilidad a la rotación de las texturas. Este método proporciona un alto rendimiento en la clasificación empleando para ello menos características y menor superficie de análisis en las imágenes que otros métodos existentes.

Muchos de los métodos existentes basados en filtros de Gabor adolecen de una serie de inconvenientes. Algunos emplean un número demasiado elevado de características para representar una textura [18], [19]. Esto conlleva un gasto computacional importante cada vez que una textura desconocida se intenta clasificar. Por ejemplo, en [19] usan 18 características para alcanzar una tasa de acierto del 92.4% y en otros métodos es necesario un número todavía mayor. También es corriente usar un área de 256x256 [20]. Aunque se consiguen buenas prestaciones en clasificación cuando se escoge un área grande, éstas, a veces, o bien no están disponibles, o no son prácticas. Por ello es preferible poder clasificar imágenes de menor resolución o tamaño.

Otro problema en la mayoría de los métodos de clasificación de texturas es su incapacidad de mantener una tasa alta de clasificación cuando la imagen ha sufrido una rotación o contiene ruido. La mayoría de los métodos requieren que las texturas que han de clasificarse tengan la misma orientación que en el conjunto

de entrenamiento. Esta es una limitación importante, puesto que el ángulo con el que se observa una textura a menudo es impredecible.

En el método aquí propuesto, se usa un nuevo enfoque basado en el filtrado de Gabor para extraer características de texturas invariantes a su rotación. Con este método se extraen tan solo cuatro características de una imagen pequeña (16x16), pero se consigue una tasa alta de clasificación y resistencia al ruido.

a) Extracción de características

Para la extracción de características, aquí se empleó un banco de filtros modificado con respecto al de la expresión (2.2). Este filtro se ha convertido en circularmente simétrico para conseguir invarianza rotacional:

$$h(x, y) = g(x, y) \cdot \exp\left(-2\pi jF\sqrt{x^2 + y^2}\right)$$

$$\text{donde } g(x, y) = \left(\frac{1}{2\pi\sigma^2}\right) \cdot \exp\left(-\frac{x^2 + y^2}{2s^2}\right)$$

En este caso no es necesario rotar las coordenadas de la gaussiana, puesto que al final tendremos simetría circular. Un ejemplo de filtro de Gabor circularmente simétrico se muestra en la Figura 2.11(c,d).

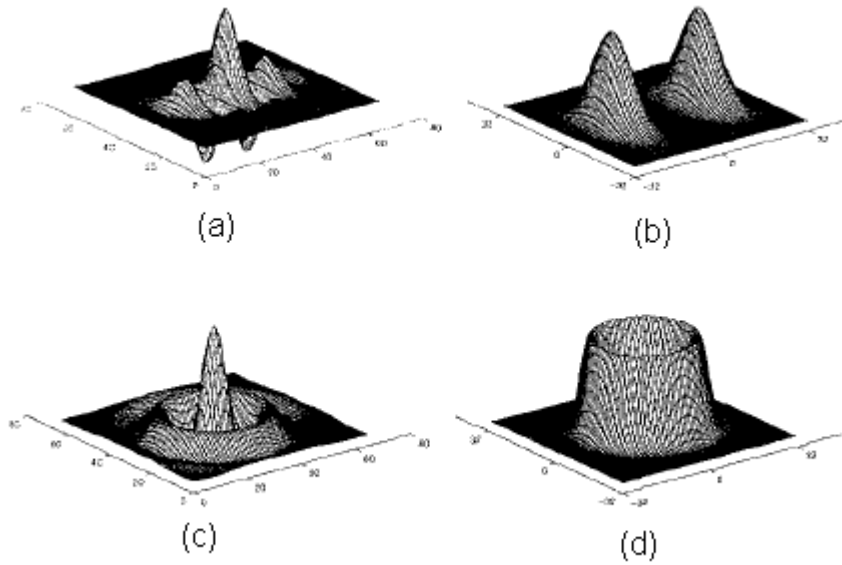


Figura 2.11: Filtro de Gabor tradicional en el dominio del espacio (a) y de la frecuencia (b). Filtro de Gabor circularmente simétrico en el dominio del espacio (c) y de la frecuencia (d). Fuente: [17].

Para extraer características de las texturas usando estos nuevos filtros circularmente simétricos emplearemos un banco de cuatro filtros, espaciados en frecuencia lo máximo posible para conseguir una máxima cobertura del espectro. Estos filtros se representan en el dominio de la frecuencia en la Figura 2.12, y se puede ver cómo el ancho de banda de cada filtro aumenta con la frecuencia. Las frecuencias más bajas no se cubren, ya que no expresan mucha información sobre la imagen (únicamente el nivel de iluminación medio). El proceso de extracción de características implica filtrar cada imagen con cada uno de los cuatro filtros y después medir cada respuesta del filtro a una imagen. En este método se usa la desviación media absoluta de la media para cada imagen filtrada:

$$f = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N |x_{i,j} - \bar{x}|$$

Donde \bar{x} es la media de la imagen filtrada y M y N son sus dimensiones. f se calcula para cada imagen filtrada para proporcionar cuatro características invariantes a la rotación.

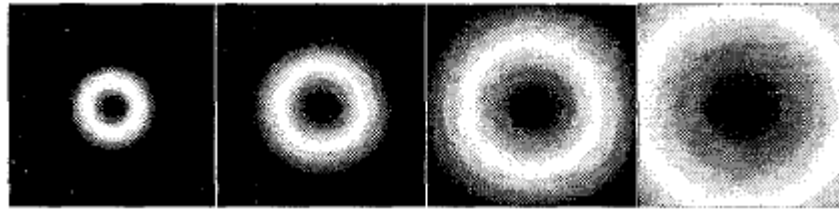


Figura 2.12: Representación en el dominio de la frecuencia del conjunto de filtros de Gabor circularmente simétricos, para imágenes de 16x16. Fuente: [17].

b) Resultados de la clasificación

En el experimento se emplearon 16 texturas de 256x256 del álbum de Brodatz [21] para probar las prestaciones de este método propuesto. Las muestras de estas texturas se exponen en la Figura 2.13. Se ha usado una imagen de muestra de cada textura para proporcionar un conjunto de entrenamiento de subimágenes de 16x16. Estas texturas se presentaron con ángulos de 0°, 30°, 45° y 60° para entrenar el algoritmo de clasificación. Se presentaron otras 7 muestras de cada textura como texturas desconocidas para la clasificación. Estas texturas se ofrecieron con ángulos de 20°, 70°, 90°, 120°, 135° y 150°. Se empleó un clasificador de mínima distancia usando la distancia de Mahalanobis [20] para llevar a cabo la clasificación, es decir, el método del ‘vecino más próximo’.

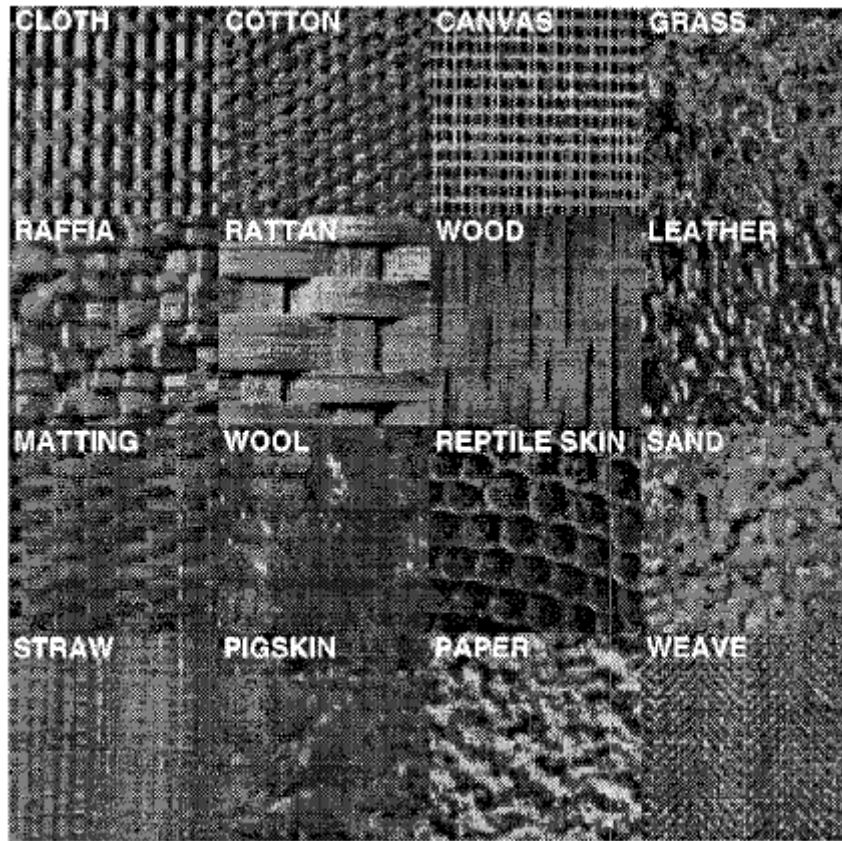


Figura 2.13: Las 16 texturas empleadas para probar el rendimiento de los filtros de Gabor circularmente simétricos. Fuente: [17].

Los filtros de Gabor circularmente simétricos obtuvieron una precisión en la clasificación del 95.4% a lo largo de las 672 muestras, como se muestra en la matriz de confusión de la Tabla 2.1. En ella se ve que la mayoría de errores de clasificación se dieron por el hecho de que las texturas “matting” (estera) y “raffia” (rafia) se confundieron con “rattan” (ratán). Esto se debe a la relación entre las frecuencias espaciales de estas texturas: rafia y estera tienen aproximadamente el doble de la frecuencia espacial que ratán y, por ello, tienen componentes importantes de la misma frecuencia que ratán. Los errores de clasificación de este tipo son de esperar cuando la base de datos es tan grande como en este caso.

c) Análisis de resultados

A diferencia de los filtros de Gabor tradicionales, estos nuevos filtros en el enfoque propuesto han demostrado ofrecer unos niveles de precisión altos. Es más, el esquema emplea significativamente menos características que la mayoría de esquemas existentes, resultando en una clasificación más sencilla y rápida. Este esquema también precisa de imágenes relativamente más pequeñas, obteniéndose un número mayor de aplicaciones con una carga computacional menor.

2.3. Conclusiones

En este capítulo se ha estudiado, con un enfoque matemático, cómo son los filtros de Gabor. Hemos visto que sirven como filtros paso banda y, con ellos, podremos extraer características frecuenciales de las texturas para facilitar su clasificación. Se estudiaron dos experimentos en los que se pretendía, por una parte cuantificar la similitud entre texturas y, por otra, clasificar texturas de forma insensible a la orientación con que éstas se presenten al sistema (mediante filtros de Gabor modificados). Este último aspecto, el de conseguir invarianza rotacional (así como podría quererse tener invarianza frente al escalado de las texturas) es algo que va más allá de nuestros objetivos por lo que, aunque digno de mención, no será objeto de estudio en el capítulo de experimentos; todas las texturas se presentarán con la misma orientación al clasificador.

Capítulo 3. Extracción de características mediante Análisis de Múltiples Variables

En el capítulo anterior vimos un método de extracción de características basado en conocimiento experto a priori; es decir, los filtros que se elegían ya estaban predefinidos, y no dependían de la base de datos en cuestión; simplemente se seleccionaban las componentes frecuenciales de las texturas intentando emular el sistema de reconocimiento del cerebro humano.

Quizás, si tenemos en cuenta la base de datos con el objeto de maximizar la probabilidad de clasificación exitosa, podrían mejorarse las prestaciones, pues parece natural pensar que una extracción de características adaptada al problema desempeñará una mejor función.

Los métodos de análisis de múltiples variables son técnicas de síntesis de la información, o reducción de la dimensión (número de variables). Es decir, ante un conjunto de datos con muchas variables, el objetivo será reducirlas a un menor número perdiendo la menor cantidad de información posible.

En este capítulo se analizarán dos grupos de métodos de extracción de características mediante aprendizaje máquina: uno de modo no supervisado y el segundo, que será de mayor interés, de modo supervisado. Estos métodos no son exclusivos del ámbito de la clasificación de texturas, sino de cualquier problema general de clasificación, por lo que en ocasiones el análisis tendrá un enfoque en el que nos abstraeremos de que las características con que trabajamos pertenecen en realidad a texturas.

3.1. Análisis de Componentes Principales

El Análisis de Componentes Principales (PCA, Principal Components Analysis) es un método de extracción de características no supervisado. Esto significa que la reducción se hará a partir de la inspección de los datos de entrada, pero no de sus etiquetas, o respuestas asociadas. Las variables extraídas con este método se denominan componentes principales, y son una combinación lineal de las variables originales [22].

3.1.1. Planteamiento del problema

Supongamos que se dispone de n imágenes con p características cada una, cuya dimensionalidad queremos reducir; éstas se disponen en forma de matriz, \mathbf{X} , de dimensiones $n \times p$, donde las columnas contienen las variables (características) y las filas, cada una de las muestras (imágenes). Supondremos que previamente hemos restado a cada variable su media, de manera que las variables de la matriz \mathbf{X} tienen media cero y su matriz de covarianzas vendrá dada por $\frac{1}{n} \mathbf{X}^T \mathbf{X}$. El problema que se intenta resolver es representar los datos con un menor número de variables, de forma que la información que se pierda sea mínima. Esto puede abordarse desde dos perspectivas equivalentes.

a) Enfoque estadístico

Si queremos representar puntos p -dimensionales en un espacio unidimensional, esto equivale a sustituir las p variables originales por una nueva variable, \tilde{x}_1 , que resuma óptimamente la información. Matemáticamente esto supone que la nueva variable debe tener máxima correlación con las originales o, dicho de otro modo, debe permitir prever las variables originales con la máxima precisión. Esto no será posible si la nueva variable toma un valor semejante en todos los elementos, y, se demuestra en [23] que, para que podamos prever con la mínima pérdida de

información los datos observados, debemos utilizar la variable de máxima variabilidad.

En la Figura 3.1 se representan los datos de entrada \mathbf{X} mediante una nube de puntos. De entre todas las posibles rectas en que se pudiesen proyectar éstos, si escogemos la que está dibujada, el nuevo conjunto de datos $\widetilde{\mathbf{X}}$, resultado de proyectar \mathbf{X} sobre la recta, presentará la máxima varianza. Esta recta es la que elegiría PCA para proyectar los datos y pasar de un espacio bidimensional a otro unidimensional. Su dirección viene dada por un vector de proyección, al que nos referiremos como \mathbf{u}_1

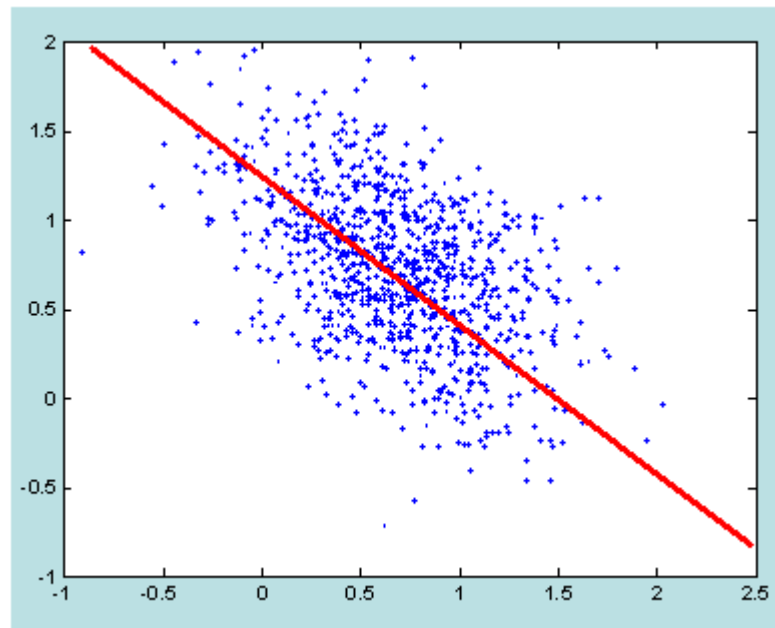


Figura 3.1: Recta que maximiza la varianza de los datos (PCA).

Este enfoque puede ampliarse para obtener la mejor proyección de los datos a un espacio bidimensional. Para ello, calcularemos el plano que mejor aproxima los puntos. El problema es equivalente a encontrar una nueva dirección definida por un vector unitario \mathbf{u}_2 . Este vector puede tomarse ortogonal a \mathbf{u}_1 sin pérdida de generalidad, y debe cumplir la condición de que la proyección de un punto sobre este eje maximice las distancias entre los puntos proyectados. Esto equivale a

encontrar una segunda variable \tilde{x}_2 , incorrelacionada con la anterior, y que tenga varianza máxima. En el caso general, la componente \tilde{x}_r ($r < p$) tendrá varianza máxima entre todas las combinaciones lineales de las p variables originales, y además deberá estar incorrelacionada con las $\tilde{x}_1, \dots, \tilde{x}_{r-1}$ previamente obtenidas.

b) Enfoque geométrico

El problema anterior puede abordarse desde un punto de vista geométrico y llegar al mismo resultado final. Volviendo a la nube de puntos de la Figura 3.1, podemos ver que los datos se distribuyen en forma de elipse, y que la recta dibujada es el eje principal de dicha elipse (en caso de que no formaran una elipse el objetivo seguiría siendo el mismo: maximizar la varianza de los datos proyectados). Al trabajar en varias dimensiones, considerando que las nubes de puntos formen elipsoides, la mejor aproximación a los datos es la proporcionada por su proyección sobre el eje mayor del elipsoide. Intuitivamente la mejor aproximación en dos dimensiones es la proyección sobre el plano de los dos ejes mayores del elipsoide y así sucesivamente. Considerar los ejes del elipsoide como vectores de proyección para extraer las nuevas variables supone pasar de variables correlacionadas a variables incorrelacionadas como veremos a continuación.

3.1.2. Cálculo de las componentes

a) Cálculo de la primera componente

La primera componente principal se define como la combinación lineal de las variables originales que tiene varianza máxima. Los valores en esta primera componente de las n imágenes se representarán por un vector $\tilde{\mathbf{x}}_1$, dado por $\tilde{\mathbf{x}}_1 = \mathbf{X}\mathbf{u}_1$.

Como las variables originales tienen media cero, lo mismo sucederá para $\tilde{\mathbf{x}}_1$. Su varianza será

$$\frac{1}{n} \tilde{\mathbf{x}}_1^T \tilde{\mathbf{x}}_1 = \frac{1}{n} \mathbf{u}_1^T \mathbf{X}^T \mathbf{X} \mathbf{u}_1 = \mathbf{u}_1^T \mathbf{C}_{xx} \mathbf{u}_1 \quad (3.1)$$

donde \mathbf{C}_{xx} es la matriz de covarianzas de las observaciones. Es evidente que podemos maximizar la varianza sin límite aumentando el módulo del vector \mathbf{u}_1 . Para que la maximización de (3.1) tenga solución debemos imponer una restricción al módulo del vector \mathbf{u}_1 , y, sin pérdida de generalidad, impondremos que $\mathbf{u}_1^T \mathbf{u}_1 = 1$. Se puede demostrar que esto es equivalente a:

$$\mathbf{C}_{xx} \mathbf{u}_1 = \lambda \mathbf{u}_1 \quad (3.2)$$

lo que implica que \mathbf{u}_1 es un autovector de la matriz \mathbf{C}_{xx} y λ su autovalor correspondiente. Para determinar cuál de los autovalores de \mathbf{C}_{xx} es la solución de (3.2) que maximiza la varianza, multiplicando por la izquierda por \mathbf{u}_1^T esta ecuación,

$$\mathbf{u}_1^T \mathbf{C}_{xx} \mathbf{u}_1 = \lambda \mathbf{u}_1^T \mathbf{u}_1 = \lambda$$

Y, por (3.1), concluimos que λ es la varianza de $\tilde{\mathbf{x}}_1$. Dado que ésta es la cantidad que queremos maximizar, λ será el mayor autovalor de la matriz de covarianzas \mathbf{C}_{xx} . Su autovector asociado, \mathbf{u}_1 , define los coeficientes de cada variable en la primera componente principal.

b) Generalización

Análogamente, puede demostrarse que el espacio de dimensión r que mejor representa los puntos viene definido por los autovectores asociados a los r mayores autovalores de \mathbf{C}_{xx} . Estas direcciones se denominan direcciones principales de los datos y las nuevas variables por ellas definidas componentes principales. En general, la matriz \mathbf{X} (y por tanto la \mathbf{C}_{xx}) tiene rango p (si $p < n$), y existen tantas componentes principales como variables, que se obtendrán

calculando los autovalores $\lambda_1, \dots, \lambda_p$, de la matriz de varianzas y covarianzas de las variables \mathbf{C}_{xx} mediante

$$|\mathbf{C}_{xx} - \lambda \mathbf{I}| = 0$$

Y sus autovectores asociados son:

$$(\mathbf{C}_{xx} - \lambda_i \mathbf{I})\mathbf{u}_i = 0$$

Los términos λ_i son reales, al ser la matriz \mathbf{C}_{xx} simétrica, y positivos, ya que \mathbf{C}_{xx} es definida positiva. Puesto que \mathbf{C}_{xx} es simétrica, si λ_i y λ_j son dos raíces distintas sus autovectores asociados son ortogonales [23].

Llamaremos $\tilde{\mathbf{X}}$ a la matriz cuyas columnas son los valores de las p componentes de las n imágenes. Estas nuevas variables están relacionadas con las originales mediante:

$$\tilde{\mathbf{X}} = \mathbf{X}\mathbf{U}$$

donde $\mathbf{U}^T \mathbf{U} = \mathbf{I}$. Calcular las componentes principales equivale a aplicar una transformación ortogonal \mathbf{U} a las variables \mathbf{X} (ejes originales) para obtener unas nuevas variables $\tilde{\mathbf{X}}$ incorrelacionadas entre sí.

Con esto concluimos el estudio de PCA. Veremos a continuación las ventajas que podría tener el usar un método supervisado para extracción de características en el espacio de entrada y, opcionalmente, en el de salida.

3.2. Algoritmos de MVA supervisados

A diferencia de PCA, la familia de métodos de Mínimos Cuadrados Parciales (PLS, Partial Least Squares) tendrá en cuenta los vectores de salida a la hora de efectuar la extracción de características.

A continuación veremos una familia de métodos de aprendizaje supervisado. Estos se conocen como PLS, OPLS y POPLS. PLS proyecta tanto \mathbf{X} como \mathbf{Y} a un

nuevo espacio, y por ello se le conoce como un modelo de factores bilineales [24]. OPLS y POPLS sólo proyectan el conjunto \mathbf{X} . En lo sucesivo supondremos, al igual que hicimos con PCA, que tanto el conjunto de variables de entrada como de salida están centrados (tienen media cero).

3.2.1. Mínimos cuadrados parciales

El método original de Mínimos cuadrados parciales (PLS, Partial Least Squares) fue desarrollado por Herman Wold en 1966. La idea principal de este método es que debemos encontrar unos cuantos autovectores de determinadas matrices de modo que se establezca una correlación alta con las variables respuesta.

En este caso, además de la proyección en el espacio de entrada $\tilde{\mathbf{X}} = \mathbf{X}\mathbf{U}$, se efectúa una proyección en el espacio de salida $\tilde{\mathbf{Y}} = \mathbf{Y}\mathbf{V}$. El objetivo de PLS consiste en buscar los vectores proyectados que maximicen la covarianza entre la entrada proyectada y los datos de salida también proyectados:

$$\mathbf{u}, \mathbf{v} = \operatorname{argmax}_{\mathbf{u}, \mathbf{v}} \frac{\mathbf{u}^T \mathbf{C}_{xy} \mathbf{v}}{\mathbf{u}^T \mathbf{u} \mathbf{v}^T \mathbf{v}}$$

Esta ecuación también puede ser expresada en función de las matrices de proyección como

$$\mathbf{U}, \mathbf{V} = \operatorname{argmax}_{\mathbf{U}, \mathbf{V}} \operatorname{Tr}\{\mathbf{U}^T \mathbf{C}_{xy} \mathbf{V}\}$$

$$\text{sueto a: } \mathbf{U}^T \mathbf{U} = \mathbf{V}^T \mathbf{V} = \mathbf{I}$$

Para hallar la solución a este problema necesitamos la descomposición en valores singulares de \mathbf{C}_{xy} (en [25] se describe de forma breve en qué consiste dicha descomposición). Al igual que con PCA, las implementaciones prácticas se basan en un esquema de dos pasos, donde los vectores de proyección \mathbf{u}_i y \mathbf{v}_i se obtienen como los vectores singulares asociados al mayor valor singular de \mathbf{C}_{xy} . Después de obtener los vectores de proyección la matriz de covarianzas cruzadas

se tienen que ‘deflactar’ (eliminar los autovectores ya calculados), y el modo de hacer esto último difiere según la variante de PLS empleada.

Por ejemplo, en PLS-SB [26] la deflación tiene lugar de forma similar a PCA, proyectando las matrices \mathbf{X} e \mathbf{Y} hacia los complementos ortogonales de las direcciones dadas por \mathbf{u}_i y \mathbf{v}_i respectivamente:

$$\mathbf{X} \leftarrow \mathbf{X}[\mathbf{I} - \mathbf{u}_i \mathbf{u}_i^T]$$

$$\mathbf{Y} \leftarrow \mathbf{Y}[\mathbf{I} - \mathbf{v}_i \mathbf{v}_i^T]$$

Este proceso equivale a deflactar la matriz de covarianzas cruzadas:

$$\mathbf{C}_{xy} \leftarrow \mathbf{C}_{xy} - \sigma_i \mathbf{u}_i \mathbf{v}_i^T$$

donde σ_i es el valor singular asociado al par $\{\mathbf{u}_i, \mathbf{v}_i\}$. Haciendo esto puede verse que PLS-SB consigue los mismos vectores de proyección que se obtendrían mediante una descomposición en valores singulares de la matriz de covarianzas originales \mathbf{C}_{xy} , y que el número máximo de proyecciones que se pueden extraer coincide con el rango de dicha matriz.

Dicho esto, damos paso ahora a distintas variantes de PLS; en especial aquéllas cuyas propiedades se traduzcan en buenas prestaciones a la hora de clasificar texturas.

3.2.2. PLS ortonormalizado

El método de Mínimos Cuadrados Parciales Ortonormalizados (OPLS, Orthogonal Partial Least Squares) fue desarrollado por Keith J. Worsley en 1998; se define mediante el siguiente problema de maximización [27]:

$$\mathbf{U} = \underset{\mathbf{U}}{\operatorname{argmax}} \operatorname{Tr} \{ \mathbf{U}^T \mathbf{C}_{xy} \mathbf{C}_{xy}^T \mathbf{U} \}$$

$$\text{con la condición: } \mathbf{U}^T \mathbf{C}_{xx} \mathbf{U} = \mathbf{I}$$

En este caso, al contrario que en PLS, sólo efectuamos una reducción dimensional en el espacio de entrada, y no en el de salida. Puesto que el caso que nos ocupa en

este Proyecto es la clasificación entre diez tipos de texturas, no estamos interesados en reducir el número de clases, sino en ser capaces de decidir correctamente a cuál de esas diez pertenece una textura que llegue al clasificador; por ello, este método parece que resultará más adecuado que PLS.

PCA vimos que era un problema de autovalores estándar; en OPLS el planteamiento de optimización también se puede transformar en un problema de autovalores generalizado:

$$\mathbf{C}_{xy} \mathbf{C}_{xy}^T \mathbf{u} = \lambda \mathbf{C}_{xx} \mathbf{u} \quad (3.3)$$

La ecuación (3.3) se puede reescribir para convertirse en un problema estándar de autovalores. Si consideramos $\mathbf{w} = \mathbf{C}_{xx}^{-1/2} \mathbf{u}$ llegamos a

$$\mathbf{C}_{xx}^{-1/2} \mathbf{C}_{xy} \mathbf{C}_{xy}^T \mathbf{C}_{xx}^{-1/2} \mathbf{w} = \lambda \mathbf{w} \quad (3.4)$$

y ahora los vectores \mathbf{w} se obtienen como autovectores de una matriz simétrica. También los vectores de proyección de OPLS se pueden obtener a partir de \mathbf{w} como $\mathbf{u} = \mathbf{C}_{xx}^{-1/2} \mathbf{w}$.

Una ventaja que posee OPLS con respecto a otros métodos de análisis multivariantes es que es óptimo, en cuanto a error cuadrático medio, para efectuar regresiones lineales múltiples de \mathbf{Y} en los datos de entrada proyectados [27]. Dicho de otro modo, aunque PLS es óptimo en el sentido de que maximiza la covarianza, OPLS es óptimo con respecto a la siguiente regla:

$$\mathbf{U} = \underset{\mathbf{U}}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{XUB}\|_F^2$$

donde $\mathbf{B} = \mathbf{X}^+ \mathbf{Y}$ es la matriz de regresión óptima, $\|\cdot\|_F$ es la norma de Frobenius de una matriz (raíz de la suma de sus elementos al cuadrado) y \mathbf{X}^+ denota la pseudoinversa de Moore-Penrose $\mathbf{X}^+ = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T$.

3.2.3. OPLS con restricciones de positividad

A OPLS se le puede imponer una restricción que permita que las soluciones obtenidas tengan una interpretación física cuando \mathbf{U} se comporte como un banco de filtros (profundizaremos más sobre esto en la Sección 3.3).

En esta variante de OPLS, que denominaremos POPLS (Positive-constrained Orthonormal Partial Least Squares), tras un desarrollo matemático [15] se llega al siguiente problema de optimización:

$$\begin{aligned} \text{maximizar: } & \text{Tr}\{(\mathbf{U}^T \mathbf{C}_{xx} \mathbf{U})^{-1} \mathbf{U}^T \mathbf{C}_{xy} \mathbf{C}_{yx} \mathbf{U}\} \\ \text{sujeto a: } & \text{Tr}\{\mathbf{U}^T \mathbf{U}\} = \mathbf{I} \\ & u_{ij} \geq 0 \end{aligned} \quad (3.5)$$

donde hemos definido las matrices de covarianzas $\mathbf{C}_{xx} = \mathbf{X}^T \mathbf{X}$, $\mathbf{C}_{xy} = \mathbf{Y}^T \mathbf{X}$ y $\mathbf{C}_{yx} = \mathbf{C}_{xy}^T$, y donde hemos hecho explícita la restricción de positividad. Esta restricción adicional (3.5) es necesaria si queremos que la solución tenga sentido físico al tomar \mathbf{U} como banco de filtros (sus valores absolutos al cuadrado).

Hay varias formas de resolver el problema que acabamos de ver. Aquí se empleará un procedimiento que consistirá en calcular iterativamente el mejor filtro \mathbf{u}_m (para la m -ésima iteración), para que no sólo garanticemos que $\mathbf{U} = [\mathbf{u}_1 \mathbf{u}_2 \dots \mathbf{u}_{n_f}]$ es el mejor banco de filtros con n_f filtros, sino que además cualquier subconjunto de bancos que consista en alguna de las primeras columnas de \mathbf{U} es también óptimo con respecto al número de filtros utilizados.

Resumiendo, el proceso consiste en las siguientes dos etapas diferenciadas:

- 1) Resolver el problema de optimización para un filtro dado por:

$$\text{maximizar: } \frac{\mathbf{u}^T \mathbf{C}_{xy} \mathbf{C}_{yx} \mathbf{u}}{\mathbf{u}^T \mathbf{C}_{xx} \mathbf{u}} \quad (3.5)$$

$$\text{sujeto a: } \mathbf{u}^T \mathbf{u} = 1 \quad (3.6)$$

$$u_i \geq 0 \quad (3.7)$$

- 2) Extraer de \mathbf{Y} la predicción obtenida del banco de filtros actual

La Tabla 3.1 resume el algoritmo de POPLS para el diseño supervisado de bancos de filtros. También hay que mencionar que, en esta implementación, el problema de maximización (3.5)-(3.7) se resolvió con la función de Matlab *fmincon*. Sin embargo, en la mayoría de las ocasiones, la convergencia de esta rutina no fue satisfactoria, haciendo necesario recurrir a una representación alternativa de \mathbf{u} basada en coordenadas hiperesféricas. La ventaja de esta representación es que la restricción (3.6) está directamente incorporada a la representación, lo que simplifica la aplicación de cualquier algoritmo de optimización [15].

Tabla 3.1: Pseudocódigo de POPLS

Entradas: \mathbf{X} , \mathbf{Y} , n_f

1. $\mathbf{C}_{xx} = \mathbf{X}^T \mathbf{X}$, $\mathbf{Y}^{(1)} = \mathbf{Y}$
2. Para cada $m = 1, \dots, n_f$
 - a. $\mathbf{C}_{xy}^{(m)} = \mathbf{X}^T \mathbf{Y}^{(m)}$; $\mathbf{C}_{yx}^{(m)} = \mathbf{C}_{xy}^{(m)T}$
 - b. Resolver (3.5)-(3.7) para obtener \mathbf{u}_m
 - c. $\mathbf{Y}^{(m+1)} = \left[\mathbf{I} - \frac{\mathbf{X}_C \mathbf{u}_m \mathbf{u}_m^T \mathbf{X}_C^T}{\mathbf{u}_m^T \mathbf{C}_{xx} \mathbf{u}_m} \right] \mathbf{Y}^{(m)}$
3. Banco de filtros de salida: $\mathbf{U}_o = [\mathbf{u}_1, \dots, \mathbf{u}_{n_f}]$

El algoritmo POPLS ya ha sido aplicado con éxito a la clasificación de géneros musicales, en un experimento, llevado a cabo en [15], en el que muestras de canciones de 30 segundos se categorizaban en uno de los once géneros posibles. Quizás, por esto, el experimento podría extenderse al caso bidimensional de la clasificación de texturas y esperar resultados similares.

3.3. Comparación de los distintos métodos

Una vez hemos estudiado distintos métodos de extracción de características cabe preguntarse cuál de los vistos será el que más nos ayude en el problema de la clasificación de texturas.

Supongamos un problema de clasificación como el ilustrado por la Figura 3.2. La posición de cada uno de los puntos mostrados representa al dato de entrada \mathbf{x} , y su color, la clase a la que pertenece (en este caso sólo hay dos clases, y_1 e y_2). Si quisiéramos efectuar una reducción de la dimensionalidad del espacio de entrada (es decir, pasar de dos dimensiones a una en este caso), ¿qué rectas de proyección elegiría cada uno de los métodos? Repasemos los conceptos vistos:

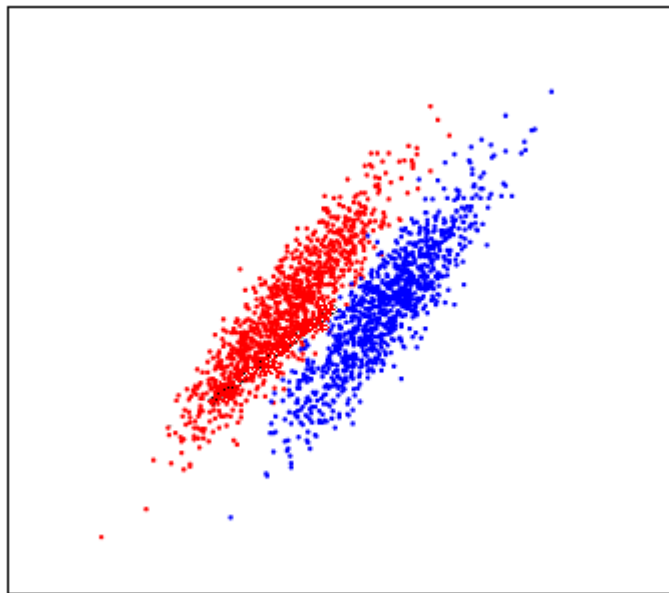


Figura 3.2: Conjunto de datos para un problema de clasificación.

- PCA busca la máxima varianza entre los datos de entrada, sin tener en cuenta la clase a la que pertenecen.
- PLS obtiene la máxima covarianza entre los datos de entrada y los de salida.

- En OPLS se busca la matriz de proyección \mathbf{U} tal que $\mathbf{U} = \underset{\mathbf{U}}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{XUB}\|_F^2$, es decir, que las salidas etiquetadas y estimadas sean lo más parecidas entre sí.

Ilustramos el funcionamiento de estos tres métodos en la Figura 3.3. Se han representado las rectas escogidas, así como la proyección de los datos de entrada sobre cada una de ellas.

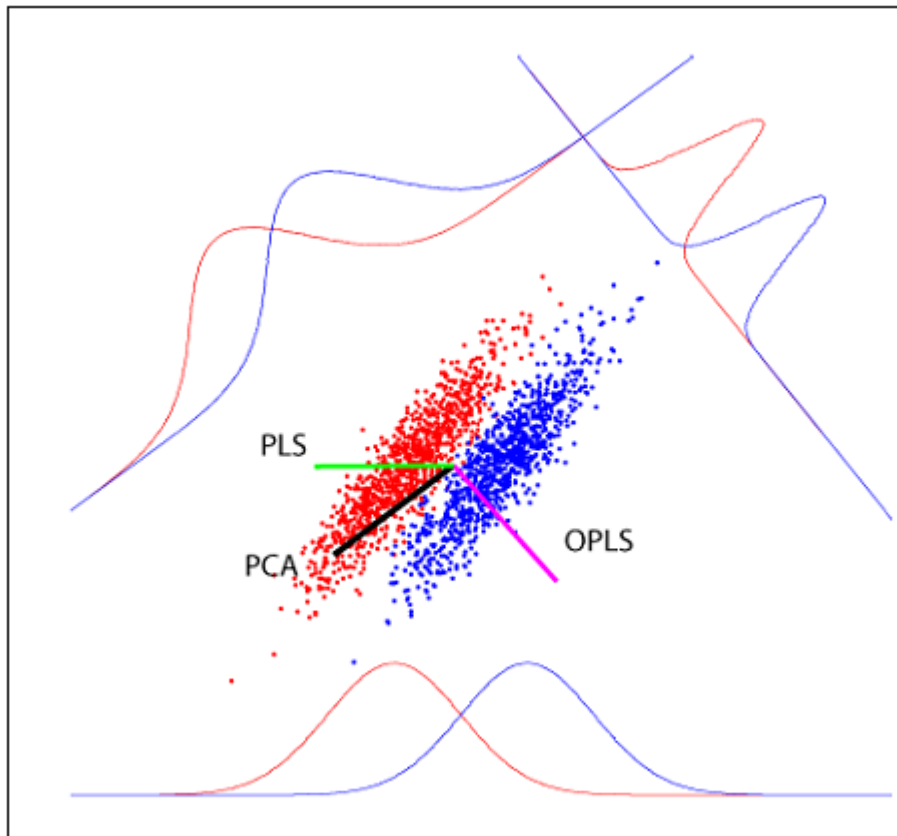


Figura 3.3: Comparativa de los PCA, PLS y OPLS. Fuente: [27].

Entonces, en un problema de clasificación, parece ser que OPLS es el que mejor separa los datos de entrada, ya que en los otros dos, el solape es mayor. Si a OPLS además se le impone la restricción de positividad (POPLS), las matrices de

proyección \mathbf{U} obtenidas tendrán un significado físico en el caso de la clasificación de texturas (aunque esto será a costa de una peor separabilidad). Dicho significado consiste en lo siguiente:

Consideremos \mathbf{X} la matriz de datos de entrada de dimensión $n \times p$, donde n es el número de texturas y p el número de variables con que se representa cada textura. En el análisis que haremos, cada una de las p variables representará el módulo al cuadrado de una componente frecuencial de la textura, por lo que cada columna de la matriz \mathbf{U} , \mathbf{u}_i al multiplicarse por una textura \mathbf{x}_j (fila de \mathbf{X}) se puede interpretar como el procesado de esta última por un filtro más la extracción de la energía resultante:

$$\mathbf{x}_j \mathbf{u}_i = x_{j1} u_{i1} + x_{j2} u_{i2} + \dots + x_{jp} u_{ip}$$

y, dado que trabajamos con módulos al cuadrado de los espectros, cada componente de \mathbf{u}_i debe ser positiva, de ahí la utilidad de la restricción de positividad de POPLS.

Por tanto, gracias a sus buenas prestaciones en cuanto a separabilidad y a su interpretación física, POPLS será el método escogido para la clasificación de texturas.

3.4. Conclusiones

Hemos analizado dos grupos principales de extracción de características: los no supervisados y los supervisados:

- Dentro de los no supervisados se analizó el método PCA, el cual busca maximizar la varianza de los datos de entrada en un nuevo espacio, donde se proyectarán éstos.
- Entre los supervisados se estudiaron los siguientes métodos:

- PLS, cuyo objetivo es maximizar la covarianza entre los datos de entrada y los de salida.
- OPLS, que intenta que las salidas etiquetadas y estimadas sean lo más parecidas posible.
- POPLS, cuyo objetivo es el mismo que el de OPLS (en la medida de lo posible), pero con la restricción de que las matrices de proyección halladas tengan signo positivo.

Todos estos métodos reducen la dimensionalidad en el espacio de entrada de los datos; adicionalmente, PLS la reduce también en el espacio de salida. Esta última característica no interesa en el problema de clasificación, por lo que no aporta ninguna utilidad adicional en el objetivo del Proyecto.

Al examinar gráficamente las proyecciones de un conjunto de datos para un problema de clasificación, se determinó que la buena separabilidad que ofrecía OPLS sería idónea para el tema que nos ocupa en este trabajo. Además, la restricción de positividad de POPLS dota de significado físico a las soluciones obtenidas, ya que, si consideramos tales soluciones como la energía de los filtros que procesan las imágenes de entrada, dicha energía solo puede tomar valores positivos.

Capítulo 4. Experimentos: Clasificación de texturas

Ya hemos visto que la extracción de características para clasificación de imágenes puede hacerse de forma fija, mediante bancos de filtros predefinidos (Capítulo 2), o bien mediante filtros elegidos dinámicamente en función de la base de datos de que dispongamos (Capítulo 3). El objetivo de este capítulo es comparar ambos métodos en un marco de trabajo común, que será la clasificación de texturas.

4.1. Elección de la base de datos

El primer paso que tenemos que dar es elegir lo que queremos clasificar, qué características deben tener las texturas que buscamos y dónde encontrarlas. Se determinó, con el objetivo de disponer de una base de datos suficientemente rica y compleja, que necesitábamos disponer de diez clases (categorías) con al menos 30 texturas por clase. Con ello deberíamos tener suficientes muestras tanto para entrenar el sistema como para evaluarlo.

En la Tabla 4.1 se listan las fuentes candidatas para nuestra base de datos, donde se analiza el número de texturas totales que contienen, el número de clases, el tipo de acceso (gratis o de pago) y comentarios que determinen su idoneidad. Los problemas que presentaron algunas de las candidatas fueron:

- Texturas poco homogéneas: no nos sirve una textura de muestra clasificada como “hierba” si ésta contiene además una segadora o una verja en la imagen [28].
- Pocas texturas por clase: para poder entrenar el clasificador necesitamos un número mínimo de muestras de entrenamiento por clase, y otro tanto

para evaluar las prestaciones. No nos sirve disponer de sólo 3 muestras dentro de una clase [29], [30], [31].

- Texturas no etiquetadas: en este caso no podemos emplear aprendizaje máquina supervisado [20].
- Calidad de las texturas innecesariamente alta: esto conllevaba dos problemas: primero, que eran de pago (y bastante costosas); y segundo, que iba a ralentizar el sistema al tener que procesar imágenes de tan alta resolución [32].

Tabla 4.1: Posibles fuentes para nuestra base de datos de texturas

Base de datos	Número de texturas	Número de clases	Acceso	Comentarios
Image After [28]	400+	20	Libre	Texturas poco homogéneas
Mayang's Free Textures [29]	1200+	60	Libre	Demasiadas clases con pocas muestras
CG Textures [33]	16000+	33 + subclases	Libre	Base de datos de texturas masiva
UWA. Texture and Colour [31]	1600+	94	Libre	Demasiadas clases con pocas muestras
Brodatz [20]	112	N.A.	De pago	Las texturas no aparecen etiquetadas
Vistex [30]	120	19	Libre	Pocas imágenes por clase
Got3D Textures [32]	3000+	≈ 15 (útiles)	De pago	Texturas de calidad excepcional

Por todo ello, se decidió escoger [33], ya que disponía de un número más que suficiente tanto de clases como de texturas por clase; además de ofrecer imágenes de calidad y ser de acceso libre.

Las texturas seleccionadas, como ya adelantamos en el primer capítulo, son las que se muestran en la Figura 4.1.



Figura 4.1: Subconjunto de la base de datos de texturas.

4.2. Preprocesado de imágenes

Una vez construida nuestra base de datos tuvimos que efectuar un conjunto de operaciones sobre las imágenes. El porqué de la elección de ciertos parámetros en el preprocesado se hizo con el objetivo, por una parte, de maximizar la tasa de aciertos en clasificación para el caso inicial de filtros de Gabor, y por otra, de conseguir un funcionamiento eficiente del algoritmo POPLS.

Dado que el número de parámetros que se hubo de tener en cuenta fue relativamente elevado, todos los experimentos que ilustren la elección de dichos parámetros se realizarán en torno al valor final del conjunto de los mismos. Adelantaremos ahora sus valores y, seguidamente, explicaremos su elección:

1. Conversión a blanco y negro.
2. Recorte a 500 x 500 píxeles.
3. Transformación al dominio frecuencial.
4. Promediado del espectro a 10 x 10 puntos.

También avanzaremos (pese a que no se verá en esta sección) que, además de los parámetros del preprocesado, el resto de condiciones en las que se hicieron los experimentos fueron:

- Banco de 16 filtros de Gabor.
- Método de validación cruzada “leave-one-out”
- Regresión lineal múltiple en el decisor.

4.2.1. Conversión a formato blanco y negro

El color de las imágenes se representa mediante un vector de tres componentes: rojo, verde y azul; dado que los filtros que hemos descrito hasta ahora son funciones que mapean un espacio bidimensional a otro unidimensional, debemos

convertir dicho vector de color a un escalar que represente la intensidad total de la imagen. Esta operación se puede realizar de manera sencilla mediante el comando `rgb2gray` de Matlab.

4.2.2 Recorte a 500x500 píxeles

Si bien este paso no es necesario cuando se extraigan las características mediante filtros de Gabor (pues los filtros pueden adaptarse al tamaño de la textura que procesan), es conveniente que todas las imágenes tengan el mismo número de puntos cuando se les aplique el método POPLS, ya que los filtros que se hallen de ésta última forma son constantes en tamaño y aplicarse a imágenes de distinta resolución requeriría un diseño complicado.

500x500 era el tamaño de la imagen más pequeña de nuestra base de datos, por lo que todas las demás se recortaron para igualarse a aquélla, como se muestra en la Figura 4.2.

La resolución original era de 800x609; vemos que al recortarla a 500x500 no se pierde información esencial en la textura.

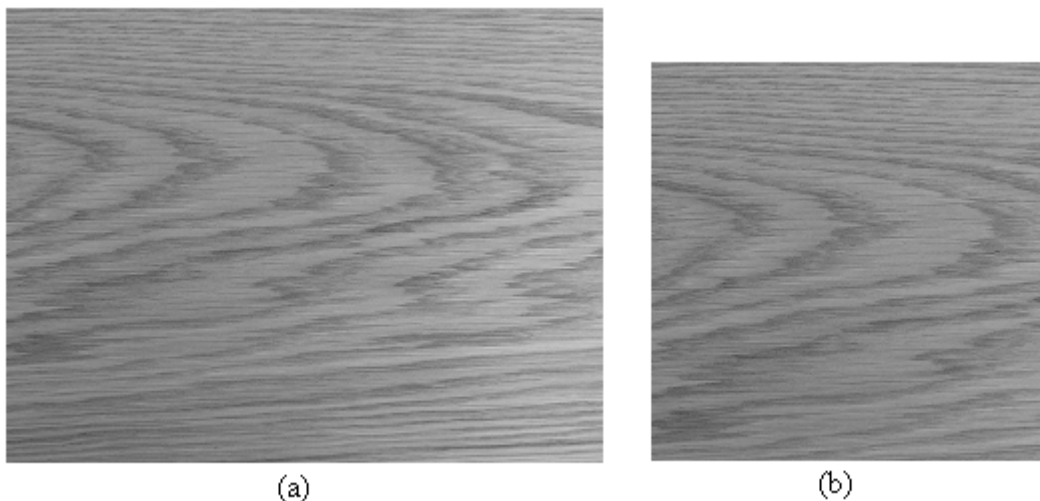


Figura 4.2: Contrachapado. (a): resolución original (800x609). (b): resolución final (500x500).

Adelantamos aquí que la extracción de características con POPLS requería que, por motivos computacionales, la textura de entrada fuese de tamaño reducido, no más de 130 puntos, y con una resolución de 500x500 se obtienen 250,000 puntos. Por tanto vamos a ver primero si el recortar la imagen para conseguir un menor número de puntos tiene impacto sobre las prestaciones del sistema de clasificación. Con los parámetros que ya definimos al principio de la sección y variando solamente el recorte de las imágenes obtenemos la gráfica mostrada en la Figura 4.3. Como se puede apreciar, cuanto más se reduce el recorte peores son las prestaciones, pues las texturas van perdiendo más y más información hasta quedar irreconocibles, incluso para el ojo humano. En la Figura 4.4 se muestra cómo sería la textura de Fig. 4.2 cuando ha sido recortada a una resolución de 20x20; se hace realmente difícil su clasificación.

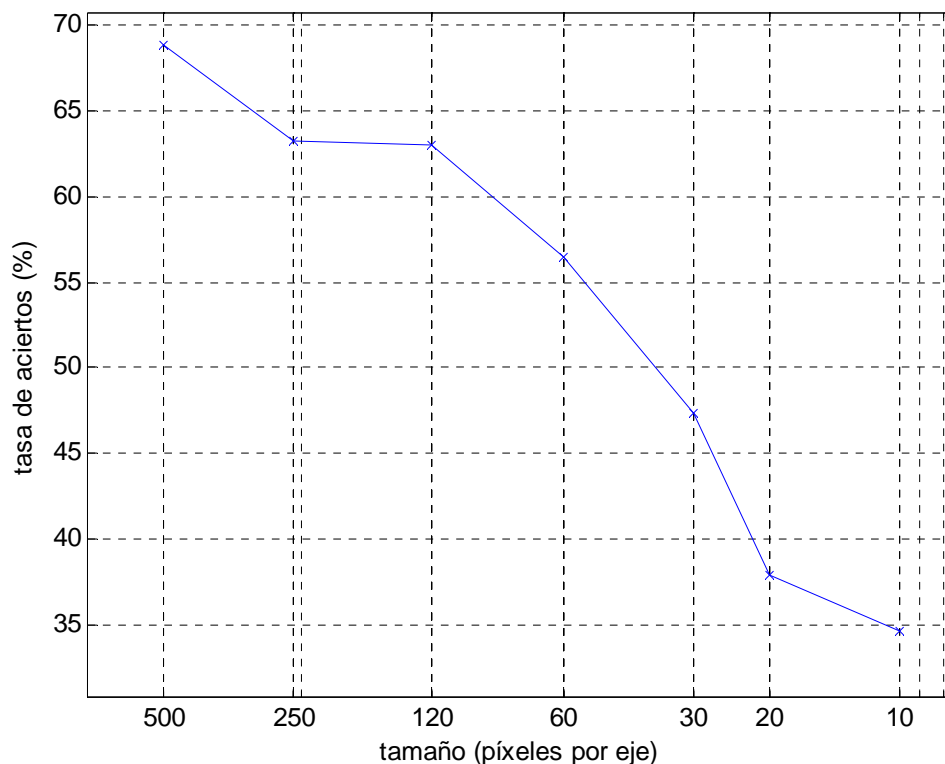


Figura 4.3: Tasa de aciertos frente a tamaño de las imágenes.



Figura 4.4: Resultado (ampliado) de recortar la textura de la Fig. 4.2 a un tamaño de 20x20.

Por todo esto, el querer compactar la información de las texturas mediante su recorte no parece buena idea, así que trabajaremos con imágenes de 500x500.

4.2.3. Transformación al dominio frecuencial

Dado que el análisis que haremos para ambos métodos de extracción de características es exclusivamente en frecuencia (como se concluyó en el Apartado 1.4.2) se deberá pasar del dominio espacial de las imágenes al dominio frecuencial mediante la DFT. Es decir, las características que extraeremos posteriormente serán un resumen de las componentes frecuenciales de las texturas. En la práctica nos quedamos con el módulo al cuadrado de la DFT, ya que ésta operación era necesaria realizarla de todos modos en el cálculo de la energía pero, en ocasiones, mostraremos sólo el módulo (y, a veces, el logaritmo del módulo) con fines ilustrativos. En la Figura 4.5 se muestra cómo es una textura en el dominio espacial y su transformación al dominio de la frecuencia; en esta última (obtenida mediante Matlab) se ha efectuado una transformación logarítmica para poder apreciar mejor las distintas componentes frecuenciales, pero en la práctica dicha transformación no se efectuó.

Puede observarse lo siguiente en el espectro de la Fig. 4.5(b):

- El espectro ha sido centrado (Sección A.1), por lo que la frecuencia cero, o valor medio, aparece representada en el centro.

- La mayor parte de la energía está en las bajas frecuencias (esto se ve todavía más acentuado si no se efectúa la transformación logarítmica).
- El espectro es prácticamente simétrico con respecto al origen, al igual que ocurre con los espectros de la Sección A.1.

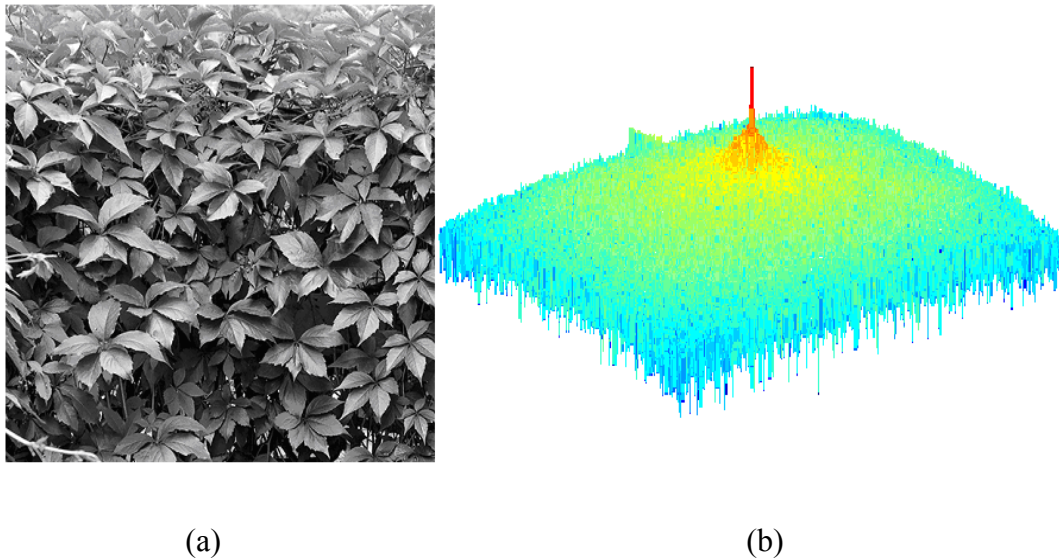


Figura 4.5: (a): Textura 'hiedra' en el dominio espacial. (b): Logaritmo de su DFT.

4.2.4. Promediado del espectro a 10 x 10 puntos

Una resolución de 500 x 500, es decir, 250,000 puntos por textura, no era problema a la hora de procesar con filtros de Gabor, ya que éstos eran fijos, estaban establecidos de antemano; no era el caso, sin embargo, con POPLS, donde los filtros debían calcularse mediante un algoritmo computacionalmente costoso. A partir de 129 puntos (los que se emplearon en el experimento de [27]) tanto el tiempo de cómputo como la memoria requerida crecían muy rápidamente. Por ello fue necesario promediar la DFT hasta alcanzar un tamaño de datos de entrada lo suficientemente pequeño.

En la Figura 4.6 se muestra el modo en que se promedia un espectro 500x500 para llegar a una nueva resolución de 50x50. De nuevo, sólo a efectos ilustrativos, las

DFTs están en escala logarítmica, ya que se ve más fácilmente de este modo, debido al menor rango de variación de los datos. Esta transformación se ve que suaviza el espectro sin apenas causar pérdida de información, cosa que no ocurría cuando se pretendía reducir la dimensionalidad mediante el recorte de la imagen.

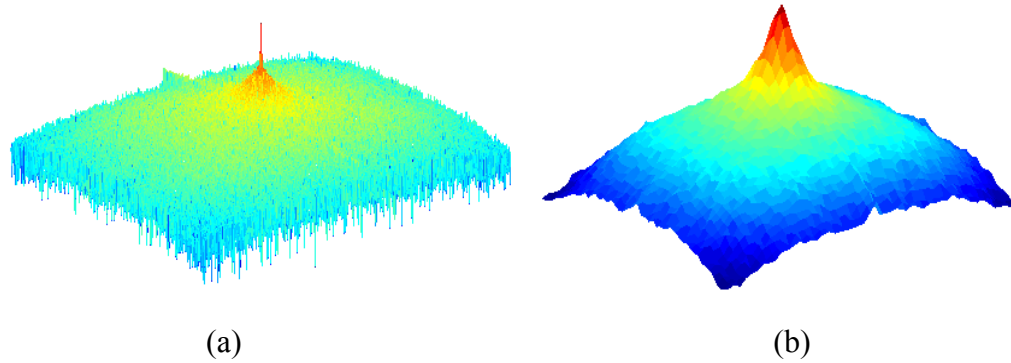


Figura 4.6: (a) Espectro de 500x500 puntos. (b) Su promediado a 50x50 puntos

La forma de promediar el espectro fue la siguiente: Cada punto del espectro promediado, $E_2(x, y)$ se calculaba como

$$E_2(x, y) = \frac{1}{s^2} \sum_{i=(x-1)s+1}^{xs} \sum_{j=(y-1)s+1}^{ys} E_1(i, j)$$

Donde E_1 es el espectro original y s es el tamaño del promediado que hay que realizar en cada uno de los ejes. Por ejemplo, en el caso de la Figura 4.6 $s = 10$, ya que se promedian regiones de 10×10 .

En la Figura 4.7 se ha dibujado una gráfica donde, a partir de texturas de 500×500 puntos se han hecho distintos promediados y evaluado sus prestaciones. La tasa de aciertos se mantiene casi constante (incluso aumenta) hasta llegar a 10×10 ; a partir de ahí, como es de esperar, la información perdida se acusa y las prestaciones empeoran. Por tanto, trabajando con texturas de 500×500 promediadas en frecuencia a 10×10 , se alcanza una tasa de aciertos satisfactoria. Nótese que pese a que trabajamos ahora con sólo 100 puntos, este no es el mismo caso que el del Apartado 4.2.2, donde la imagen se recortaba en el dominio espacial a 10×10 y luego se tomaba su DFT.

La principal razón de que las prestaciones mejoren al ir bajando la resolución hasta llegar a 10x10 puntos no es intuitiva, pero se debe a que un promediado de los filtros de Gabor hace que, en este caso, el punto de frecuencia cero tome valores no nulos si en el espectro original valía cero. Esto será importante cuando expliquemos el diseño de los filtros de Gabor, y se puede apreciar si se observa la Figura 4.8. Aunque a simple vista es difícil de ver, en el filtro de 500x500 el origen de frecuencias vale cero, y en el de 50x50 ya no; además, aunque en el primero el origen de frecuencias no fuese exactamente cero, la presencia de tantas otras componentes frecuenciales (hay un total de 250,000) hace que el peso de dicha frecuencia cero sea insignificante; no es así para el de 50x50, donde el número total de componentes frecuenciales es mucho menor, y el valor de frecuencia cero no se ve eclipsado en tal medida como antes.

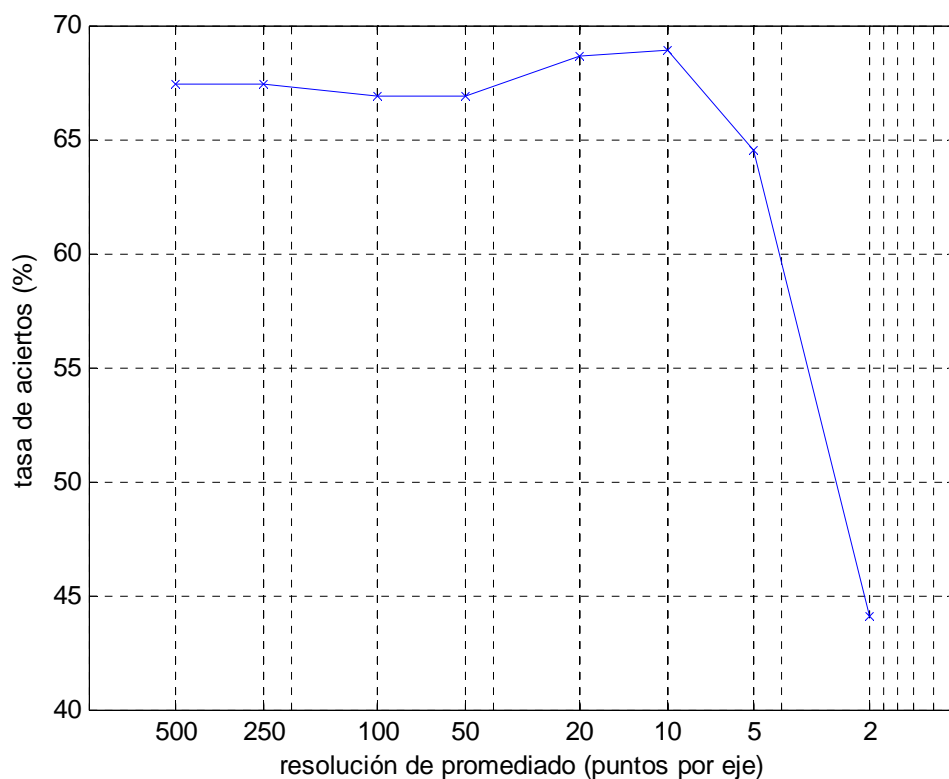


Figura 4.7: Tasa de aciertos en clasificación frente a resolución del promediado de texturas.

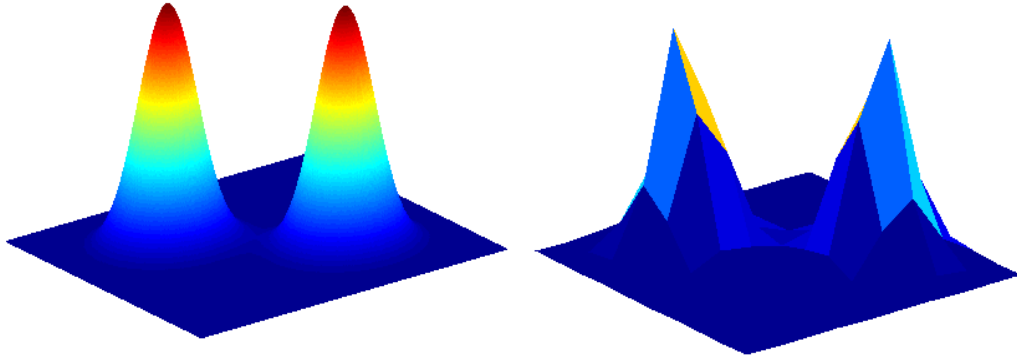


Figura 4.8: Resultado de promediar el espectro de un filtro de Gabor de 500x500 a 10x10.

4.3. Descripción del clasificador empleado

Una vez efectuada la extracción de características, para poder clasificar las muestras observadas es preciso, como se vio en el Apartado 1.3.4 establecer fronteras de decisión. La forma más sencilla de frontera en el espacio multidimensional es el hiperplano, y éste se consigue mediante regresión lineal múltiple de la siguiente forma:

1. Partimos del conjunto de datos de entrada $\tilde{\mathbf{X}}$ y de salida \mathbf{Y} . Hay que encontrar la matriz de pesos \mathbf{B} tal que $\hat{\mathbf{Y}} = \tilde{\mathbf{X}}\mathbf{B}$, donde ahora $\hat{\mathbf{Y}}$ es la salida estimada, que se desea que sea una buena aproximación de \mathbf{Y} .
2. \mathbf{B} suele calcularse como $\mathbf{B} = \tilde{\mathbf{X}}^+ \mathbf{Y} = \left(\tilde{\mathbf{X}}^T \tilde{\mathbf{X}} \right)^{-1} \tilde{\mathbf{X}}^T \mathbf{Y}$, pero eso sería cuando el número de texturas por clase fuese el mismo para todas ellas. Para ponderar igual a todas las clases debemos emplear la expresión

$$\mathbf{B} = \left[\frac{\tilde{\mathbf{X}}_1}{\#\mathbf{n}_1}; \frac{\tilde{\mathbf{X}}_2}{\#\mathbf{n}_2}; \dots \frac{\tilde{\mathbf{X}}_{10}}{\#\mathbf{n}_{10}} \right]^+ \left[\frac{\mathbf{Y}_1}{\#\mathbf{n}_1}; \frac{\mathbf{Y}_2}{\#\mathbf{n}_2}; \dots \frac{\mathbf{Y}_{10}}{\#\mathbf{n}_{10}} \right], \text{ donde } \tilde{\mathbf{X}}_i, \mathbf{Y}_i \text{ son el}$$

subconjunto de $\tilde{\mathbf{X}}$, \mathbf{Y} que se refieren a texturas de la clase i , y $\#n_i$ es el número de texturas de la clase i .

- En el caso de que sea necesario ajustar el valor medio de los datos, es decir, cuando la matriz de datos $\tilde{\mathbf{X}}$ no está centrada, habrá que transformar ésta primero en $\tilde{\mathbf{X}}_e = [\mathbf{1} \ \tilde{\mathbf{X}}]$, donde $\mathbf{1}$ representa una columna de unos, y repetir el paso 2 para $\tilde{\mathbf{X}}_e = \tilde{\mathbf{X}}$.
3. Una vez obtenido \mathbf{B} , cada textura nueva que llegue al sistema se evalúa como $\hat{\mathbf{y}} = \tilde{\mathbf{x}}\mathbf{B}$, donde $\hat{\mathbf{y}}$ será un vector fila cuyo elemento máximo representará la clase decidida.
- Ejemplo: para una muestra j que llega al decisor $\hat{\mathbf{y}}_j = \tilde{\mathbf{x}}_j\mathbf{B} = [0.1, 0.6, -0.2, 0.4, 0.2, -0.1, 0, 0.4, 0.3, 0.1]$ el mayor elemento es el segundo (0.6), por lo que se decidirá que la muestra pertenece a la clase 2.

Si se está haciendo una validación cruzada, para probar las prestaciones, las muestras que lleguen al decisor $\tilde{\mathbf{x}}$ irán acompañadas de su etiqueta \mathbf{y} correspondiente. Entonces mediante una comparación de \mathbf{y} con $\hat{\mathbf{y}}$ se verá si la muestra ha sido clasificada correctamente o no. En el caso del ejemplo anterior, si $\mathbf{y}_j = [0, 0, 1, 0, 0, 0, 0, 0, 0, 0]$ significa que la muestra j pertenecía en realidad a la clase 3, por lo que erraríamos en su clasificación.

El método de validación cruzada que se emplea es el denominado “leave-one-out”, donde el sistema se entrena tantas veces como número de muestras tengamos, siempre usando para dicho entrenamiento todas las imágenes menos una, que será la que se evalúe en cada iteración. Una vez hecho esto podemos evaluar las prestaciones construyendo una matriz de confusión, que tendrá la estructura que se muestra en la Tabla 4.2. Las filas indican la clase de la imagen que hay que clasificar y las columnas, las clases que se han decidido. Todos los elementos que caigan fuera de la diagonal principal se consideran errores.

Tabla 4.2: Ejemplo de matriz de confusión.

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
C1	10									
C2		8		1	1					
C3			9			1				
C4				10						
C5		2	4		4					
C6						9		1		
C7							10			
C8								10		
C9							2		8	
C10								1		9

Para calcular la tasa de aciertos sólo hay que sumar los elementos de la diagonal principal de la matriz de confusión y dividir entre el número total de imágenes.

4.4. Extracción de características mediante filtros de Gabor

Una vez hemos explicado la secuencia de pasos que suceden antes y después de la extracción de características, para poder situar ésta en un contexto completo, podemos empezar a comparar los dos métodos bajo estudio. El banco de filtros fijo (filtros de Gabor) o el obtenido mediante aprendizaje máquina supervisado (POPLS). Empezaremos con los primeros.

4.4.1. Cobertura espectral

Para la elección de un banco de filtros de Gabor se recomienda que la suma de todos ellos cubra la totalidad del espectro excepto las componentes frecuenciales más bajas ([17]). Esto es así porque las más bajas frecuencias están relacionadas con el nivel de iluminación medio de la textura, y no interesa, por ejemplo, diferenciar una textura ‘hierba iluminada’ de otra ‘hierba poco iluminada’. No obstante, en las Tablas 4.3(a,b) (tomadas en una fase temprana del análisis) se ve que, con nuestra base de datos, el tomar en cuenta las bajas frecuencias conlleva una mejora de las prestaciones. Esto es así porque, en la elección que tomamos a la hora de clasificar, ‘nieve’ y ‘cielo’ son texturas que tienen contornos similares, y el mejor modo de discriminarlas es mediante el nivel de intensidad (el cielo es ligeramente más oscuro que la nieve, como se ve en la Fig. 4.9). Por ello, es preferible cubrir todo el espectro, pues se consigue una mejor tasa de clasificación.

Tabla 4.3: Matrices de confusión para clasificación de texturas basada en características obtenidas mediante banco de filtros de Gabor. (a): Al no incluir las bajas frecuencias. (b): Al incluirlas.

	Cielo	Nieve
Cielo	37	0
Nieve	24	4

(a)

	Cielo	Nieve
Cielo	29	8
Nieve	2	32

(b)

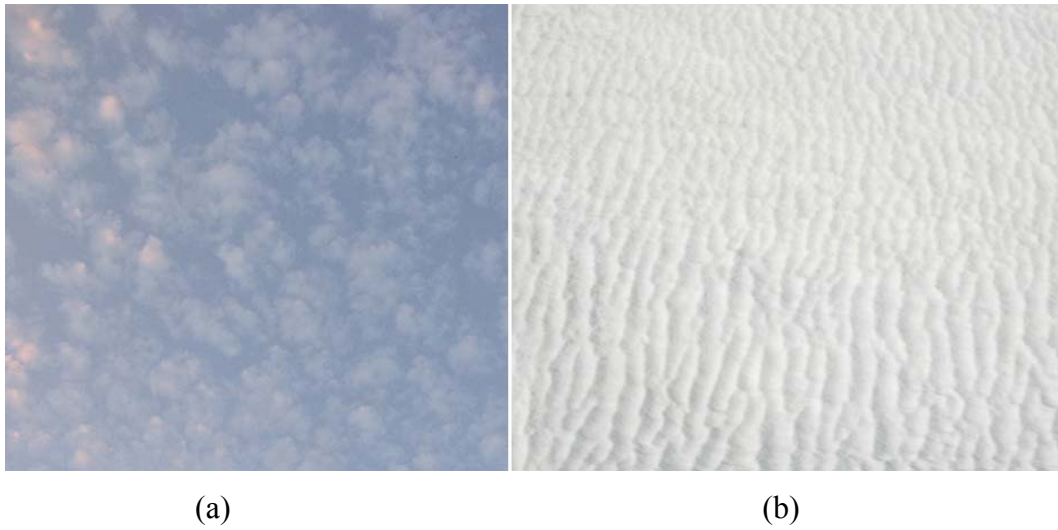


Figura 4.9: (a) Textura 'cielo'. (b) Textura 'nieve'.

De este hecho nos percatamos, como habíamos adelantado antes, tras analizar la gráfica de la Figura 4.7, y ver que la tasa de aciertos aumenta al disminuir la resolución mediante un promediado. En realidad estas mejores prestaciones podían haberse dado para la resolución 500x500 si desde un primer momento algunos filtros de Gabor hubiesen incluido la frecuencia cero.

4.4.2. Elección de características tras el filtrado

Una vez que se ha procesado una textura por uno de los filtros del banco se obtiene la imagen filtrada, de la que hay que extraer información en forma de uno o varios números. Dicha información podía ser, entre otros indicadores:

- La media.
- La desviación típica.
- La energía.

Pese a que en el experimento del Apartado 2.2.1 se utilizó la media y la desviación típica de la imagen filtrada, en nuestro caso bastó con elegir la energía para obtener resultados similares. Según el Teorema de Parseval [34], la energía de una imagen se puede calcular tanto en el dominio espacial como en el frecuencial, de la siguiente manera:

$$\sum_{x=1}^N \sum_{y=1}^N |f(x, y)|^2 = \frac{1}{N^2} \sum_{u=1}^N \sum_{v=1}^N |\hat{f}(u, v)|^2$$

donde N es el número de puntos de cada eje de la imagen, tanto en el espacio, como en frecuencia. Por ello, podemos resumir la imagen filtrada sin más que sumar todas sus componentes al cuadrado. Con esto, al tener un banco de n_f filtros, obtendríamos n_f características extraídas por textura (Fig. 4.10).

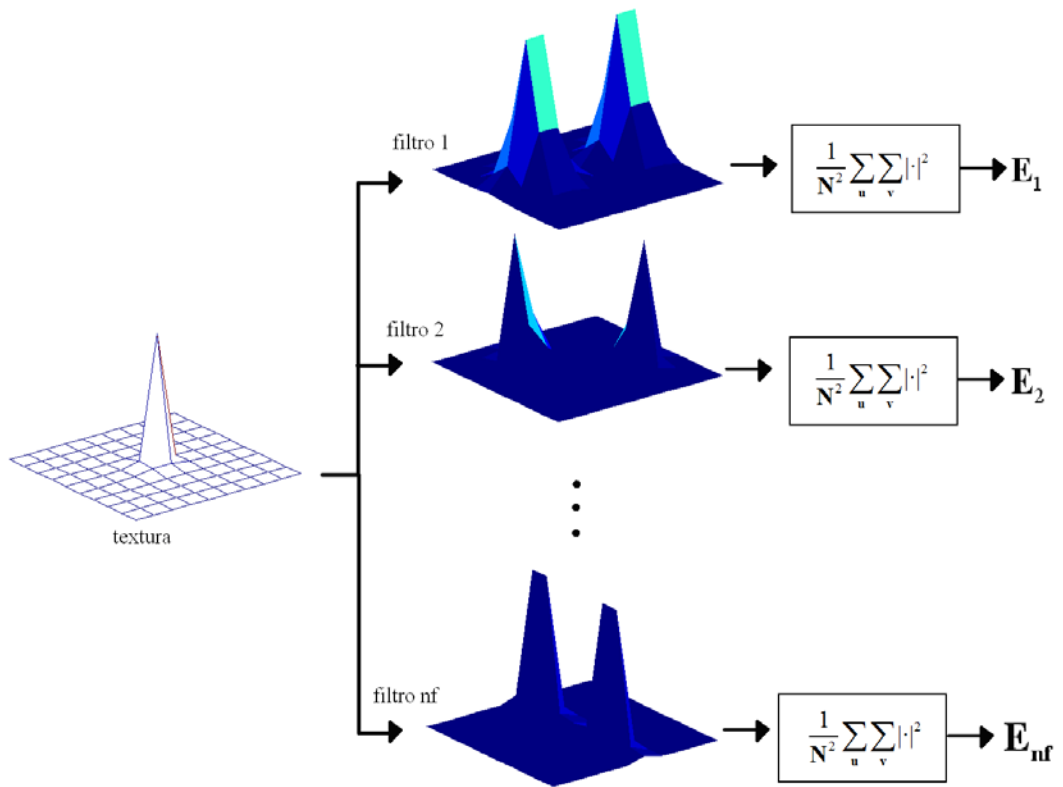


Figura 4.10: Esquema de extracción de características de una textura mediante bancos de filtros de Gabor.

4.4.3. Elección del número de filtros

El tamaño del banco de filtros de Gabor se eligió de forma empírica; se suele preferir el menor número de filtros que obtenga las mayores prestaciones aunque,

en este caso, decidimos elegir un valor mayor, para asegurarnos de que habíamos alcanzado holgadamente la zona de convergencia; los pasos dados fueron:

1. Se crea un banco con un número elevado de filtros.
2. Por cada uno de los filtros del banco:
 - a. Realizar la clasificación empleando sólo uno de ellos
 - b. Almacenar el filtro que maximice la tasa de aciertos.
3. Empleando el filtro previamente elegido y, además, cada uno de los restantes:
 - a. Realizar la clasificación empleando esos dos filtros
 - b. Almacenar el segundo filtro que maximice la tasa de aciertos.
4. Seguir almacenando los filtros que produzcan mejores resultados
5. Una vez se llegue al punto donde el introducir nuevos filtros no conlleve a mejoras en las prestaciones podemos detenernos.

Así, como se ve en la Figura 4.11, las prestaciones aumentan con el número de filtros hasta llegar, más o menos, a los 12. De todos modos, nosotros escogimos 16, para no tener que elegir un subconjunto en todos los experimentos, puesto que la representación de dicha gráfica no implica que con un banco de 12 filtros se consiga la tasa máxima de aciertos, sino que son los 12 mejores filtros de un banco de 16.

Para realizar esto es necesario disponer de un conjunto de validación y, por ello, se empleó el método “leave-one-out” del que hablamos antes.

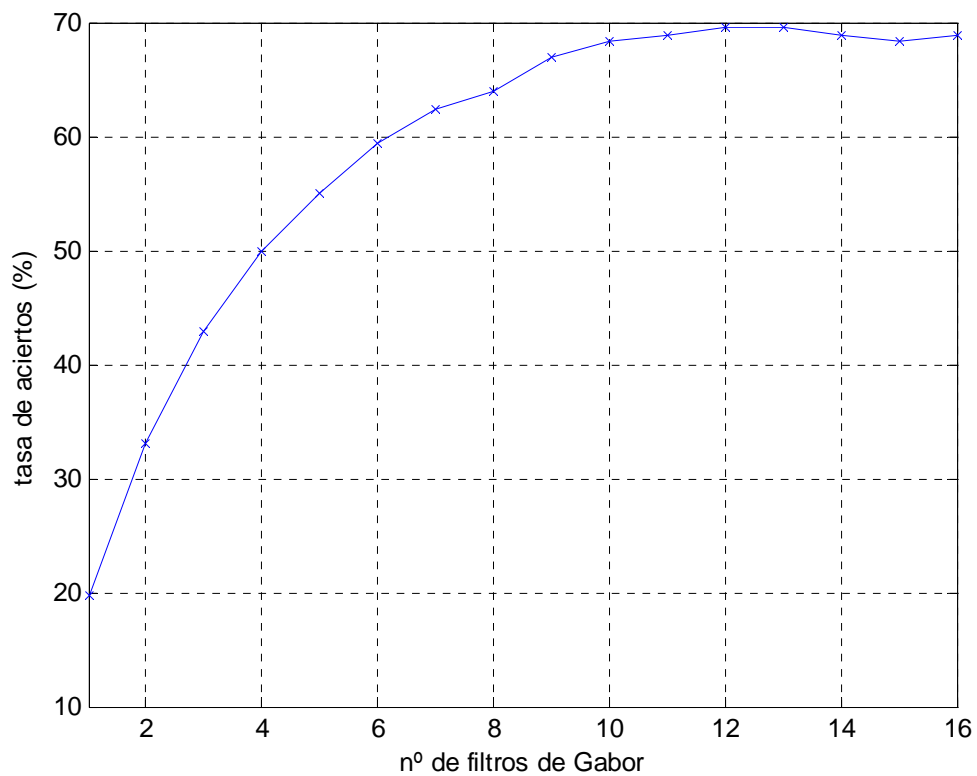


Figura 4.11: Tasa de aciertos mediante validación cruzada en función del número de filtros de Gabor empleados.

El banco total de los filtros de Gabor empleados se muestra en la Figura 4.12. Se emplearon 4 escalados y 4 orientaciones distintas. Nótese que estos filtros de Gabor, al estar a una resolución tan baja (10x10) son representaciones burdas de lo que deberían ser funciones gaussianas. También se puede apreciar el motivo de que se logre la convergencia antes de usar los 16 filtros, pues algunos de ellos se parecen entre sí.

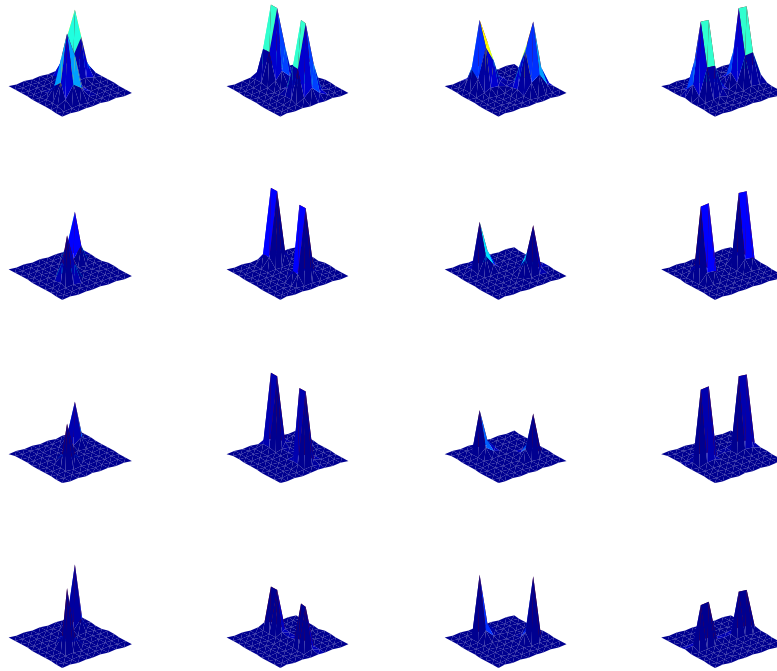


Figura 4.12: Banco de 16 filtros de Gabor (resolución de 10x10).

La matriz de confusión cuando se emplean 16 filtros se muestra en la Tabla 4.4. Obsérvese que las tasas de aciertos difieren entre unas clases y otras. Por ejemplo, grietas y contrachapado tienen una tasa de errores muy por encima de la media. Esto puede en parte ser debido a que no se está utilizando como información el color de las texturas; y así, por ejemplo, una imagen de contrachapado en blanco y negro puede confundirse con olas del agua de mar.

Tabla 4.4: Matriz de confusión para 16 filtros de Gabor. Tasa De aciertos = 68..93%

	Corteza	Ladrillo	Grietas	Hierba	Grava	Hiedra	Contrachapado	Cielo	Nieve	Agua
Corteza	26	0	0	2	0	0	0	1	1	0
Ladrillo	3	32	0	1	0	0	0	4	2	0
Grietas	0	0	7	1	2	0	1	2	13	5
Hierba	0	0	1	33	1	0	0	2	0	0
Grava	0	0	0	2	17	11	0	0	0	0
Hiedra	0	0	0	0	6	25	0	0	0	0
Contrachapado	2	1	0	0	0	0	12	3	4	10
Cielo	0	0	0	0	0	0	0	30	7	0
Nieve	1	0	0	0	0	1	0	1	32	0
Agua	0	1	4	0	0	0	2	5	2	19

4.5. Extracción de características mediante POPLS

En esta ocasión evaluaremos las prestaciones de los filtros generados automáticamente, en función de nuestra base de datos; tal y como se explicó en el Apartado 3.2.3, los filtros obtenidos de esta manera admiten una interpretación física. Conforme vayamos extrayendo características analizaremos dichas interpretaciones.

4.5.1. Algoritmo de extracción

Reiterando lo ya visto en la Tabla 3.1 el algoritmo que extrae las n_f mejores características de cada textura se detalla a continuación. En esta ocasión, tanto \mathbf{X} como \mathbf{Y} han tenido que centrarse (restarse su media), por lo que trabajaremos con sus versiones centradas \mathbf{X}_C e \mathbf{Y}_C :

1. $\mathbf{C}_{xx} = \mathbf{X}_C^T \mathbf{X}_C$, $\mathbf{Y}_C^{(1)} = \mathbf{Y}_C$
2. Para cada $m = 1, \dots, n_f$
 - a. $\mathbf{C}_{xy}^{(m)} = \mathbf{X}_C^T \mathbf{Y}_C^{(m)}$; $\mathbf{C}_{yx}^{(m)} = \mathbf{C}_{xy}^{(m)T}$

b. Resolver (3.5)-(3.7) para obtener \mathbf{u}_m

$$c. \mathbf{Y}^{(m+1)} = \left[\mathbf{I} - \frac{\mathbf{X}_c \mathbf{u}_m \mathbf{u}_m^T \mathbf{X}_c^T}{\mathbf{u}_m^T \mathbf{C}_{xx} \mathbf{u}_m} \right] \mathbf{Y}^{(m)}$$

3. Banco de filtros de salida: $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_{n_f}]$

donde ahora:

- \mathbf{X}_c : matriz que es la versión centrada de \mathbf{X} con tantas filas como imágenes y 100 columnas (resultado de desarrollar el espectro al cuadrado de la imagen 10x10 por columnas).
- \mathbf{Y}_c : matriz que es la versión centrada de \mathbf{Y} con tantas filas como texturas y 10 columnas (todas a cero excepto una, que será un uno correspondiente a la clase a la que pertenece).
- \mathbf{U} : Matriz de reducción de dimensionalidad, con 100 filas y n_f columnas, tal que $\tilde{\mathbf{X}} = \mathbf{X}\mathbf{U}$.
- $\tilde{\mathbf{X}}$: Matriz de datos de dimensionalidad reducida; con tantas filas como imágenes pero sólo n_f columnas.

De modo que, a diferencia del caso de filtros de Gabor, ahora la extracción de características se obtiene mediante la multiplicación por una matriz \mathbf{U} , con el mismo efecto que el de extraer la energía mediante el diagrama de la Figura 4.8. Cada columna de \mathbf{U} tiene 100 elementos, pero éstos se pueden redimensionar (de la misma forma que cada textura en \mathbf{X} se representó mediante un vector de 100 elementos redimensionando una matriz 10x10) en forma de matriz 10x10.

Como hemos dicho, en esta ocasión los espectros de las imágenes tuvieron que centrarse, de lo contrario no se cumplirían condiciones como $\mathbf{C}_{xx} = \mathbf{X}_c^T \mathbf{X}_c$; de todos modos esto no ocasiona la pérdida de interpretación física, ya que en [27] se explica que esto es sólo una forma equivalente de plantear el mismo problema.

También hay que señalar que, de entre todos los pasos que debía efectuar el algoritmo, el de resolver (3.5)-(3.7) era el que más tiempo de cómputo consumía. Esto es así porque se trata de un algoritmo de búsqueda (resuelto con la función `fmincon` de Matlab), y requiere un valor inicial a partir del cual se efectúa; la eficacia de dicha búsqueda depende en gran medida de este valor inicial, que no podía sino tomarse aleatorio. Para mejorar su eficacia, se efectuó la búsqueda 10 veces, con 10 valores iniciales distintos, multiplicando también por 10 el tiempo de cómputo.

4.5.2. Análisis del primer filtro obtenido

Si sólo elegimos un filtro para la clasificación, es decir, cada textura se representa mediante un solo número, en este caso \mathbf{U} es un vector 100×1 . La representación de este filtro se muestra en la Figura 4.13, y su matriz de confusión, en la Tabla 4.5.

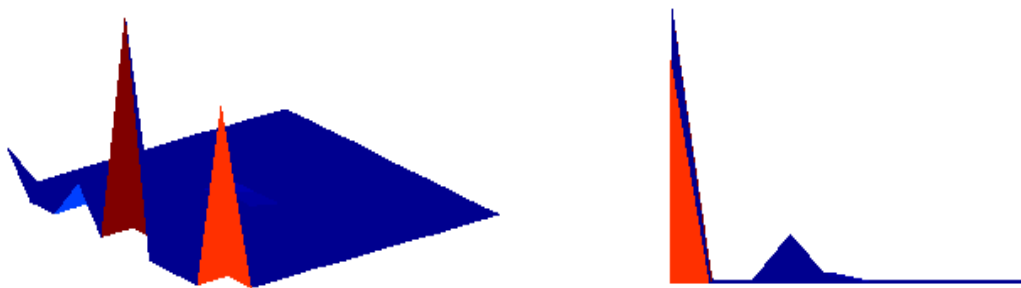


Figura 4.13: (a): Primer filtro obtenido para la clasificación mediante POPLS. (b): Vista lateral del mismo filtro.

Tabla 4.5: Matriz de confusión para un solo filtro obtenido con POPLS. Tasa De aciertos = 21.6%.

	Corteza	Ladrillo	Grietas	Hierba	Grava	Hiedra	Contrachapado	Cielo	Nieve	Agua
Corteza	0	0	0	18	0	8	0	4	0	0
Ladrillo	0	0	0	6	0	8	0	24	4	0
Grietas	0	0	0	7	0	2	0	10	12	0
Hierba	0	0	0	17	0	14	0	5	1	0
Grava	0	0	0	18	1	7	0	0	4	0
Hiedra	0	0	0	16	0	13	0	2	0	0
Contrachapado	0	0	0	0	0	7	0	18	7	0
Cielo	0	0	0	0	0	5	0	30	2	0
Nieve	0	0	0	2	0	0	0	21	12	0
Agua	0	0	0	1	0	4	0	22	6	0

En la Figura 4.13 vemos entonces que las características que mejor discriminan las imágenes son un conjunto de altas frecuencias en la dirección horizontal (y, en menor medida, algunas bajas frecuencias). Observando la Tabla 4.5 vemos que esto hace que las texturas se clasifiquen, a grandes rasgos, dentro de dos grupos principales (Fig. 4.14):

- Hierba o hiedra (texturas de altas frecuencias).
- Cielo o nieve (texturas de bajas frecuencias).

La razón de que las demás texturas no se clasifiquen bien es debido a que la primera característica no tiene en cuenta la orientación y, por tanto, si llega una textura con orientación dominante (o simplemente que no entra dentro del conjunto anterior) se clasificará dentro del grupo al que más se asemeje:

- Corteza de árbol y grava se clasifican como hierba/hiedra (altas frecuencias).
- Ladrillo, contrachapado, grietas y agua se clasifican como cielo/nieve (bajas frecuencias).

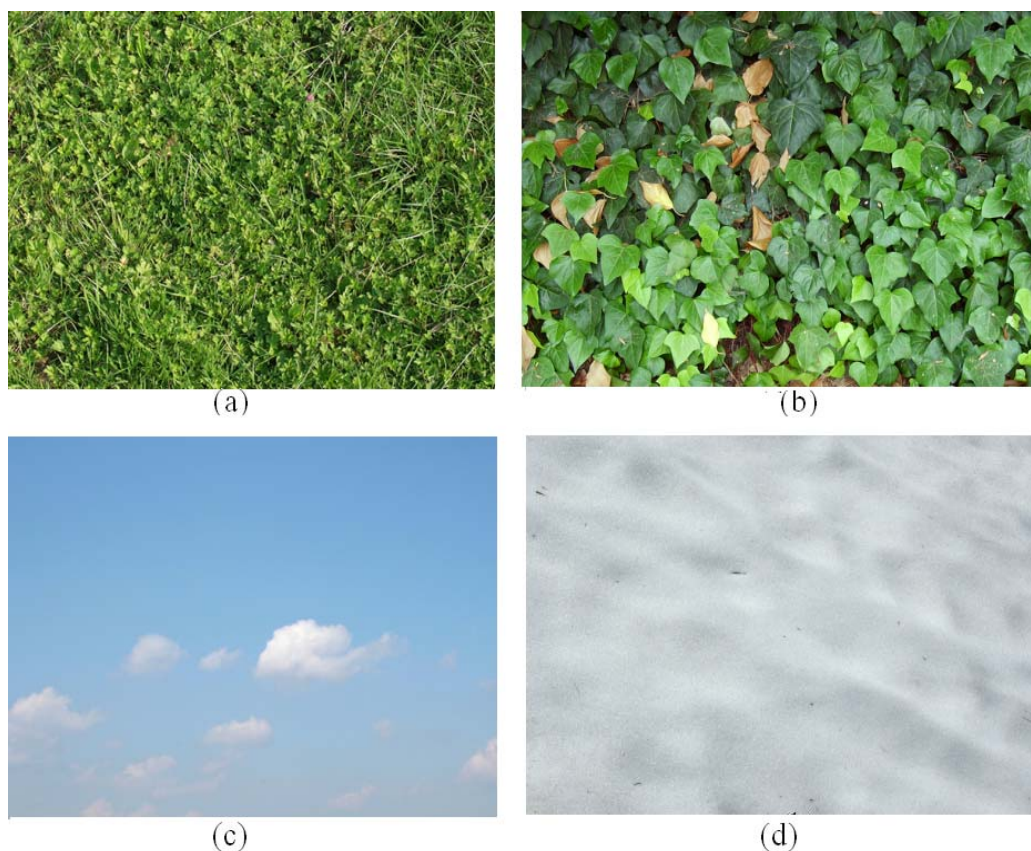


Figura 4.14: Texturas de ejemplo. (a): Hierba. (b): Hiedra. (c): Cielo. (d): Nieve

Al trabajar sólo con un filtro, las decisiones son muy burdas, basándose tan solo en si las frecuencias que predominan en las texturas son altas o bajas; al ir eligiendo más filtros, los criterios de evaluación se refinarán y, con ello, aumentará la probabilidad de acierto.

4.5.3. Evolución de las prestaciones con el número de filtros

A medida que se van obteniendo más filtros, la tasa de aciertos en clasificación aumenta. En la Figura 4.15 se puede ver el progreso que experimenta. Al igual que para el caso de Gabor, elegir 16 filtros es más que suficiente.

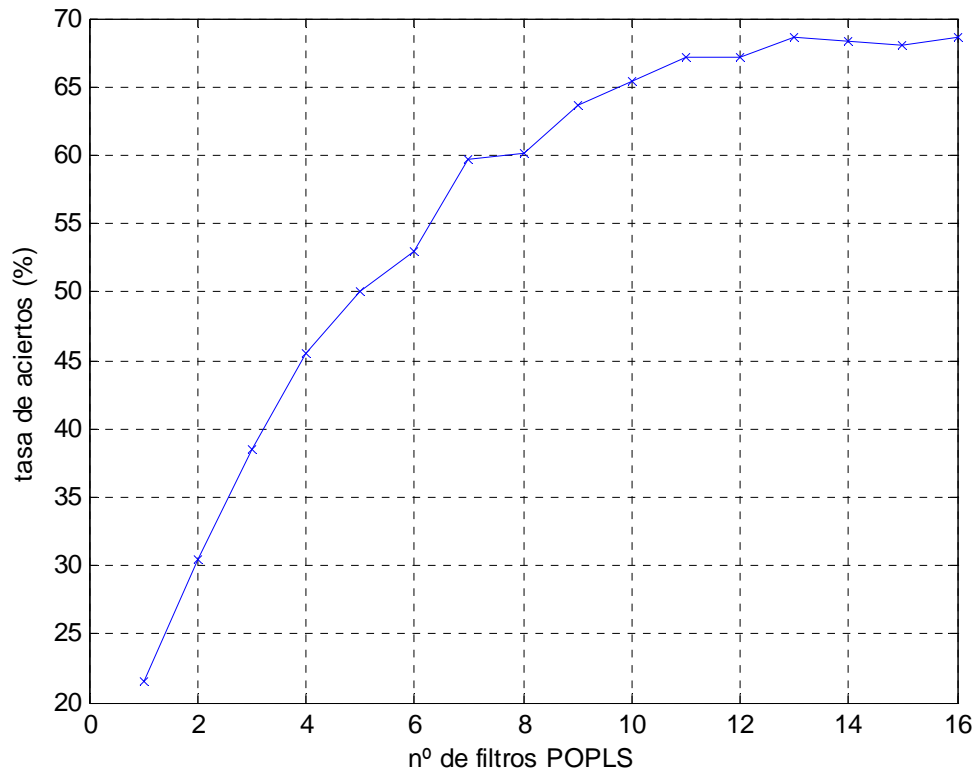


Figura 4.15: Tasa de aciertos en clasificación frente al número de filtros obtenidos con POPLS.

En la Figura 4.16 se muestran los 16 primeros filtros obtenidos mediante POPLS. Como se ve, en cada iteración se va extrayendo distintas partes del espectro. Por ejemplo, el segundo filtro calculado representa la frecuencia cero o valor medio de la imagen (que, como ya habíamos concluido, es una característica importante para la base de datos específica con que trabajamos). También, en la Tabla 4.6, se puede ver cómo es la matriz de confusión para la clasificación con 16 filtros; es casi idéntica a la obtenida con filtros de Gabor, y presenta el mismo problema de que algunas de las clases tienen una tasa de aciertos más baja.

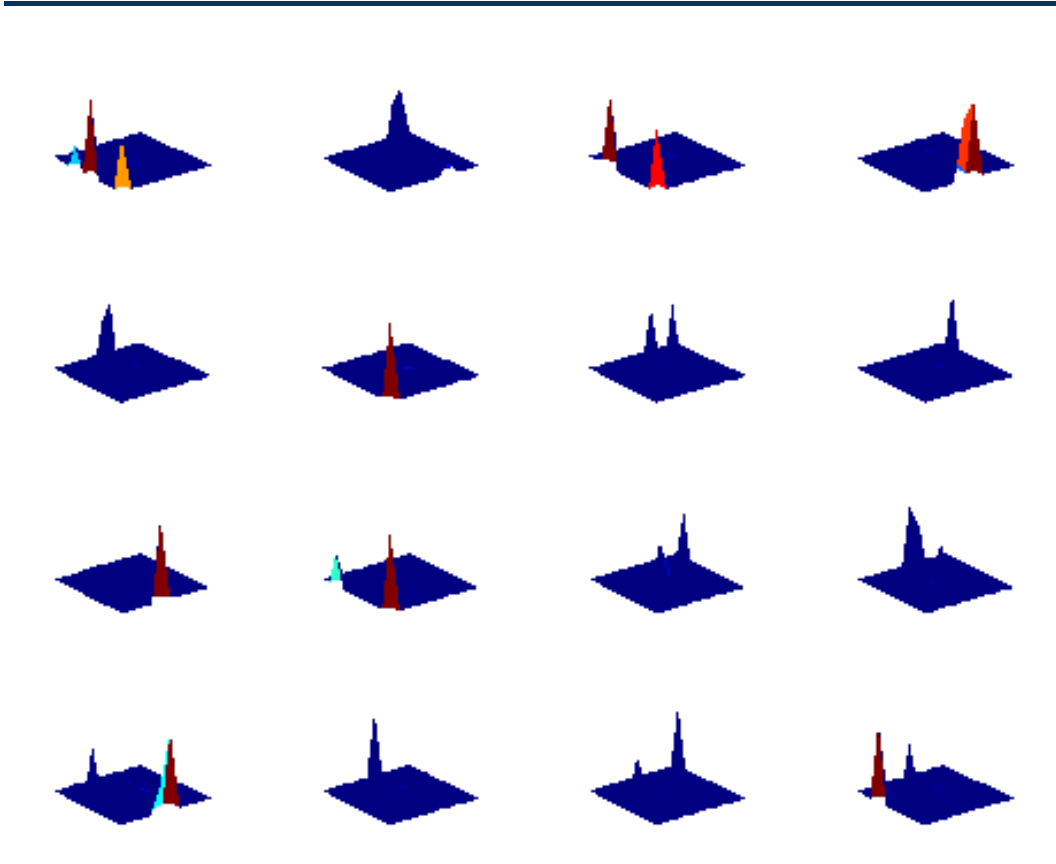


Figura 4.16: 16 primeros filtros obtenidos con POPLS

Tabla 4.6: Matriz de confusión para 16 filtros obtenidos con POPLS. Tasa De aciertos = 68.6%

	Corteza	Ladrillo	Grietas	Hierba	Grava	Hiedra	Contrachapado	Cielo	Nieve	Agua
Corteza	26	0	0	2	0	1	0	0	1	0
Ladrillo	1	30	1	1	0	0	2	5	2	0
Grietas	0	0	10	1	2	1	0	2	13	2
Hierba	1	0	0	31	1	0	0	4	0	0
Grava	0	0	1	1	17	11	0	0	0	0
Hiedra	0	0	0	0	5	26	0	0	0	0
Contrachapado	0	0	4	0	0	0	10	7	4	7
Cielo	0	0	0	0	0	0	0	30	7	0
Nieve	1	0	0	0	0	0	0	1	33	0
Agua	0	0	4	0	0	0	1	6	3	19

4.6. Comparativa entre ambos métodos

Analizando la figura 4.17 vemos que el progreso que sigue POPLS es similar al de los filtros de Gabor, si bien éste último suele quedar ligeramente por encima. Esto significa que POPLS no es capaz de aumentar la tasa de aciertos (en validación cruzada “leave-one-out”) al ritmo que lo hace el banco de filtros de Gabor (aunque la sigue de cerca, y finalmente la termina alcanzando). No obstante, la curva de Gabor es demasiado optimista, ya que los filtros se fueron eligiendo precisamente con la condición de maximizar la tasa de aciertos (además de que elegimos a propósito incluir las componentes de bajas frecuencias), por lo que dicha curva podría considerarse casi un máximo teórico para el caso de filtros de Gabor. De todos modos, esperábamos que POPLS fuese capaz de superar sus prestaciones.

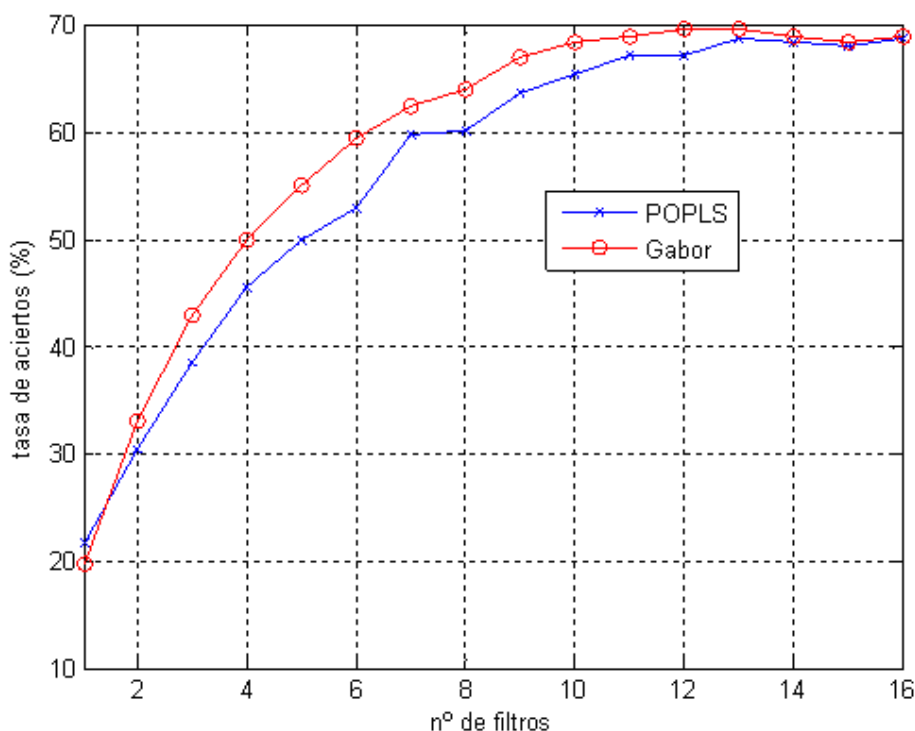


Figura 4.17: Tasa de aciertos de clasificación en validación cruzada frente al número de filtros obtenidos con POPLS y Gabor.

Para comprobar si algo fue mal a la hora de calcular los filtros mediante POPLS, se realizará otro experimento: en vez de validar mediante “leave-one-out”, dividimos la base de datos en dos grupos; el 60% de las imágenes se usarán en el entrenamiento, y el 40% restante, en la validación. Si el algoritmo hace lo que debe, entonces POPLS debería siempre hallar el mínimo de $\|\mathbf{Y} - \hat{\mathbf{Y}}\|_F^2$, como se explicó en el Apartado 3.2.2. Los resultados se pueden apreciar en la Figura 4.18. Como se ve, POPLS no consigue hallar el mínimo de la función; incluso este valor varía de una ejecución a otra, como se puede ver en las dos pruebas realizadas. Esto nos lleva a pensar que hay problemas de convergencia en el método de búsqueda fmincon de Matlab.

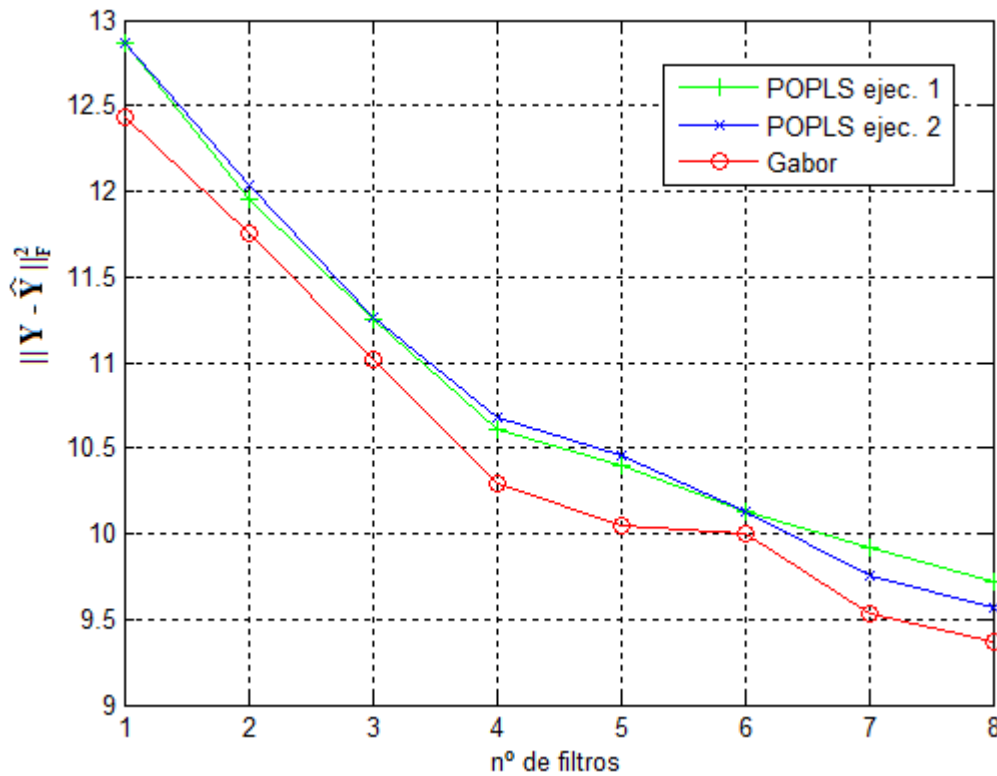


Figura 4.18: $\|\mathbf{Y} - \hat{\mathbf{Y}}\|_F^2$ en el conjunto de entrenamiento frente al número de filtros obtenidos con POPLS y Gabor.

Para comprobar que, efectivamente, el cálculo de los filtros difiere de una ejecución a otra, en la Figura 4.19 vemos de nuevo cómo serían los 16 filtros calculados para la misma imagen una segunda vez; obsérvese que muchos de ellos son radicalmente distintos. Si bien es cierto que no siempre difieren tanto los filtros hallados, se ha escogido una ejecución donde dichos filtros fueran lo menos parecidos posible a los de la Figura 4.16. Por esta razón no podemos superar a la curva de filtros de Gabor, ya que el algoritmo de optimización no logra converger bien.

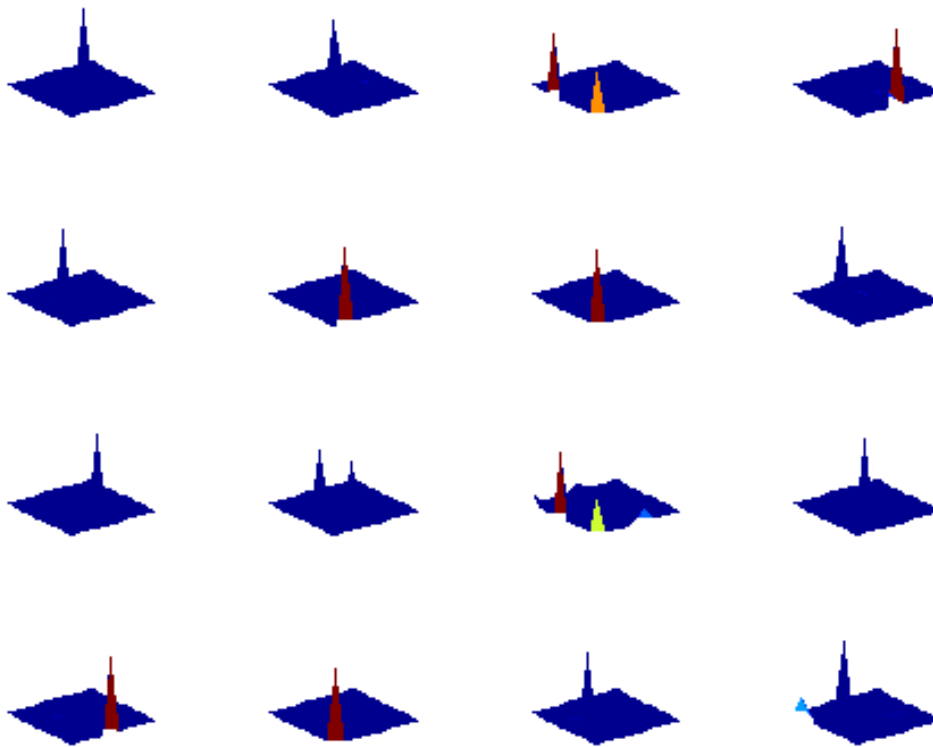


Figura 4.19: 16 primeros filtros obtenidos con POPLS (nueva ejecución).

4.7. Conclusiones

Tras crear un sistema de clasificación adecuado a la extracción de características tanto para filtros de Gabor como para POPLS (dicho sistema incluía elementos en común como el preprocesado y el decisor) se procedió a comparar ambos métodos. Se comprobó que el primero tiene ligeramente mejores prestaciones, aunque éstas fueron debidas seguramente a una convergencia no satisfactoria en el algoritmo POPLS. De todos modos los filtros hallados con este último método tienen la ventaja de que se ofrece una solución visual que define cuáles son las bandas frecuenciales más importantes a la hora de clasificar texturas, como ya vimos en el Apartado 4.5.2. Quizás con un algoritmo de búsqueda más potente que fmincon se obtendrían mejores resultados.

Independientemente, las matrices de confusión, en el mejor de los casos para ambos métodos, mostraron que hay clases que se clasifican mejor que otras, por lo que quizás en este problema específico hubiese hecho falta tener en cuenta características adicionales, como podría haber sido el color de las texturas.

Capítulo 5. Conclusiones y líneas futuras

5.1. Conclusiones

La clasificación de texturas es, a día de hoy, un campo con multitud de aplicaciones, y éstas se dan en los ámbitos de la biomedicina, la industria o la teledetección. Para poder clasificar texturas de manera exitosa es necesario disponer de un sistema de extracción de características que obtenga los rasgos más discriminatorios a la hora de clasificar. Al tener las texturas propiedades periódicas, esta extracción se puede realizar mediante el filtrado selectivo en frecuencia de la imagen para, a partir de la imagen procesada por un banco de filtros, obtener la energía resultante de cada una de ellas; de modo que extraeríamos tantas características como filtros tenga el banco.

Los filtros que procesen las imágenes deben elegirse de modo que extraigan únicamente la información (la combinación lineal de frecuencias) más relevante a la hora de clasificar, y para ello se optó por comparar dos bancos de filtros diferentes:

- Filtros fijos: que serán funciones de Gabor, por imitación al proceso de reconocimiento de texturas usado por el cerebro humano. Dichas funciones son sinusoides multiplicadas por gaussianas y, en el dominio de la frecuencia, se comportan como filtros paso banda, que seleccionan frecuencias espaciales en torno a una central.
- Filtros adaptados a la base de datos: éstos se obtendrán mediante un algoritmo de reducción de datos, denominado POPLS, cuyo objetivo es la minimización del error cuadrático residual. Dicho algoritmo ya había sido aplicado en la clasificación de géneros musicales y se pretendió extrapolarlo al caso bidimensional con texturas.

A la hora de comparar las prestaciones de ambos métodos hubo que dar los siguientes pasos, con el objeto de situar a uno y otro en las mismas condiciones:

- Crear una base de datos de texturas común: ésta debe tener un número suficiente de clases, así como un número suficiente de imágenes por clase. Las texturas se eligieron de forma que fuesen fácilmente clasificables por personas; todas ellas se daban en la naturaleza (ninguna fue generada por ordenador) y se intentó, en la medida de lo posible, que no contuviesen elementos que no fueran propios de cada clase (por ejemplo, en las texturas de “agua de mar” no incluir aquéllas donde salten a la vista grandes peces o barcos).
- Adecuar las texturas a las fases posteriores, lo que se conoce como preprocesado de las imágenes. Éste consistió en lo siguiente:
 - Convertirlas a blanco y negro, para quedarnos exclusivamente con el valor de la intensidad de las imágenes.
 - Recortarlas a un tamaño de 500x500 píxeles, para simplificar el problema, al disponer de todas las imágenes de tamaño constante.
 - Transformación al dominio de la frecuencia, pues las características periódicas de las texturas propician su estudio en dicho dominio.
 - Promediado del espectro a 10x10 puntos. Este paso fue necesario como primera reducción de variables del conjunto de entrada, pues el método POPLS no admitía una dimensión inicial demasiado elevada.
- Implementar un clasificador adecuado. Esta es la última fase, la que se sitúa después de la extracción de características, y tiene como objetivo conseguir una adecuada generalización de las clases. Se optó por el modelo más sencillo, el de clasificación mediante regresión lineal múltiple.

Una vez construido el sistema de clasificación se procedió a comparar ambos métodos. Mediante un proceso de validación cruzada se compararon las tasas de aciertos frente al número de filtros empleados. Aunque las curvas de ambas

fueron parecidas, las de los filtros de Gabor quedaron ligeramente por encima, lo que le confirió unas mejores prestaciones. De todos modos, esto fue probablemente debido a que la función encomendada a la convergencia del método de POPLS no fue satisfactoria, pues no se logró la optimización teórica que se pretendía. Asimismo la curva de Gabor era demasiado optimista, pues los filtros evaluados se elegían como aquellos que, dentro de un banco con un número elevado de filtros, obtenían las mejores prestaciones

En cualquier caso, los filtros obtenidos con POPLS se interpretaron físicamente y dieron una idea de cuáles podrían ser las componentes frecuenciales más importantes a la hora de clasificar texturas (para el caso de bases de datos específicas).

5.2. Líneas futuras de investigación

En este trabajo se ha analizado un primer enfoque a los métodos supervisados de extracción de características para la clasificación de texturas. Como ya hemos dicho, el principal problema que se tuvo en la implementación del algoritmo de POPLS fue encontrar la solución a un problema de optimización, además de que el tiempo requerido para ello resultó excesivo. Aunque se redujo considerablemente el tamaño de la entrada al extractor de características se estima que, con una entrada aún menor, este tiempo mejore. Una reducción posible del tamaño de entrada podría ser el emplear la mitad del espectro, y no completo, para a partir de éste efectuar la extracción de características, pues los espectros de imágenes tienen simetría con respecto al origen.

Por supuesto, otra solución consistiría en encontrar una función computacionalmente más eficiente, para que no sólo el tiempo de cómputo se reduzca, sino que además las soluciones así halladas sean realmente óptimas.

Independientemente, otra mejora posible para la extracción de características, tanto mediante filtros de Gabor como mediante POPLS, sería tomar otras adicionales, como por ejemplo el color, ya que algunas de las clases, al ser convertidas a blanco y negro dificultan su clasificación incluso para el ojo humano.

Como última línea de investigación también cabría añadir la posibilidad de clasificar texturas de manera invariante a la rotación o al escalado con que éstas se presenten, ya que en muchas ocasiones dichos parámetros no son rasgos que las definan.

Apéndice A. Conceptos básicos del procesamiento digital de imágenes

En este apéndice se revisará lo más básico del procesamiento de imágenes (especialmente su tratamiento frecuencial), que será fundamental para todo el Proyecto, pues todas las texturas serán procesadas en el dominio de la frecuencia por un banco de filtros.

A.1. Representación de imágenes en el dominio de la frecuencia

Las imágenes se representan de forma matemática como una matriz de píxeles; a cada coordenada (x,y) se le asocia un número, que representa su intensidad (o un vector, en el caso de imágenes en color, aunque nos centraremos en el caso de imágenes monocromáticas). Una vez tenemos esta representación podemos tratarla como una señal en tiempo (espacio en este caso) discreto y procesarla de la forma habitual. Una de las transformaciones que podemos efectuar sobre dicha imagen es la Transformada Discreta de Fourier (DFT) en dos dimensiones (2D).

Las expresiones de la transformada de Fourier y su inversa son:

$$\hat{f}(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \exp\left(-2j\pi\left(\frac{\mu x}{M} + \frac{vy}{N}\right)\right)$$
$$f(x,y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} \hat{f}(u,v) \exp\left(2j\pi\left(\frac{\mu x}{M} + \frac{vy}{N}\right)\right)$$

Donde x e y son las componentes espaciales, y u y v son las componentes frecuenciales correspondientes a x y a y , respectivamente.

También podemos visualizar la DFT; en esta ocasión la nueva imagen va a tomar valores complejos, por lo que deberemos descomponerla en dos partes: módulo

(también llamado espectro) y fase. Veamos en la Figura A.1 un ejemplo de una imagen y su correspondiente espectro:

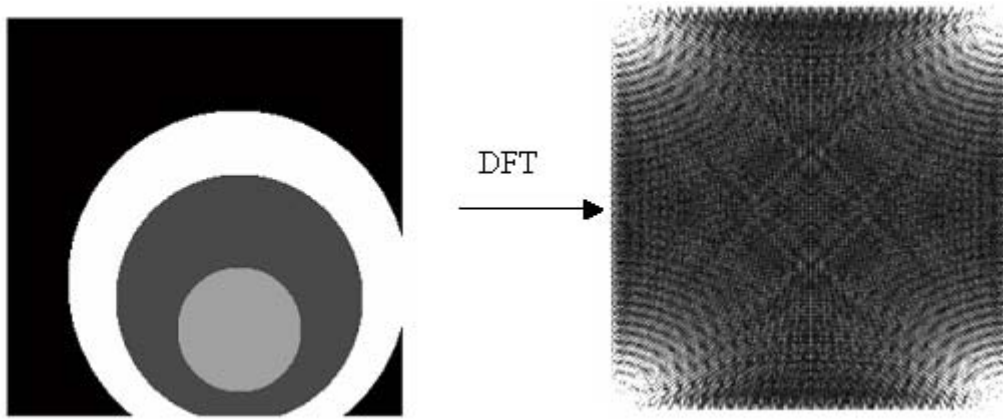


Figura A.1: Imagen y su transformada de Fourier. Fuente: [11].

Es habitual, sin embargo, representar el espectro centrado, de modo que lo que antes eran las esquinas de la transformada pasan a ser el centro de la imagen. Esto se consigue haciendo uso de la siguiente propiedad de la DFT:

$$T.F\left[f(x,y)(-1)^{x+y}\right] = \hat{f}(u-M/2, v-N/2)$$

Por tanto, para lograr el desplazamiento deseado, deberemos multiplicar por $(-1)^{x+y}$ la imagen original, como en la Figura A.2. En la imagen resultante de esta figura hay que notar la siguiente correspondencia: los puntos cercanos al centro representan las bajas frecuencias de la imagen original (es decir, variaciones suaves de intensidad), mientras que los puntos más alejados representan las altas frecuencias (variaciones abruptas de intensidad). A partir de ahora asumiremos siempre esta traslación y al ‘espectro centrado’ lo llamaremos simplemente ‘espectro’.

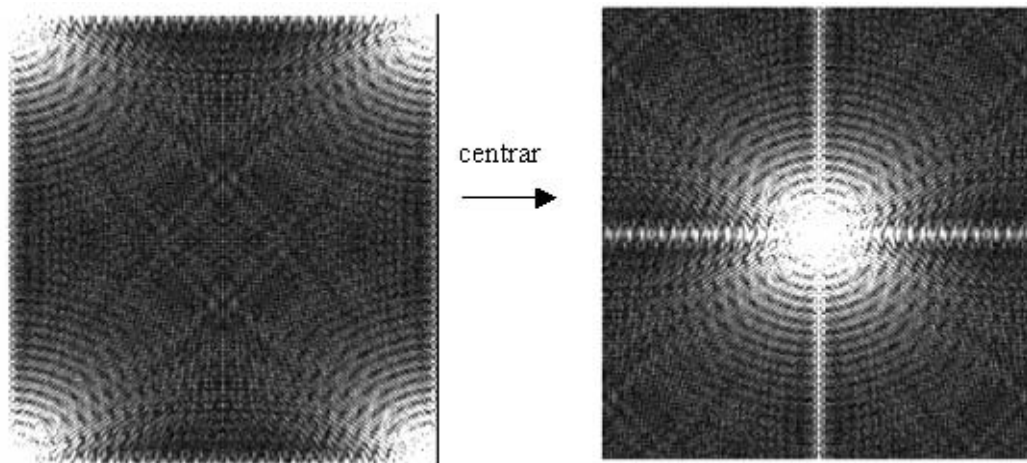


Figura A.2: Al multiplicar la imagen de la Figura A.1 por un factor $(-1)^{x+y}$ su espectro resultante queda centrado: las bajas frecuencias se representan en la parte central. Fuente: [11].

A.2. Filtrado en el dominio de la frecuencia

A.2.1. Generalidades

Se entiende por ‘filtrar’ el procesar una imagen original (imagen de entrada) mediante una máscara o filtro para obtener una imagen respuesta (imagen de salida). La utilidad del filtrado es doble: por un lado nos sirve para extraer información de una imagen (tal como su nivel medio de intensidad, o los cambios abruptos de ésta), y por otro, para eliminar efectos indeseados en la imagen (como puede ser el ruido sinusoidal).

Se puede filtrar en el dominio del espacio, mediante la convolución entre la imagen original y su máscara, o en el de la frecuencia, que es mucho más sencillo, pues sólo se necesita multiplicar el espectro de la imagen con el de la máscara. Por ello nos centraremos exclusivamente en el segundo método.

Filtrar en el dominio de la frecuencia requiere la siguiente secuencia de pasos:

1. Multiplicar la imagen de entrada por $(-1)^{x+y}$, como ya habíamos dicho, para centrar el espectro.
2. Computar $\hat{f}(u, v)$, la DFT de la imagen del paso 1.
3. Multiplicar $\hat{f}(u, v)$ por una ‘función de filtro’ $\hat{h}(u, v)$.
4. Computar la Transformada Inversa, IDFT, resultante del paso 3.
5. Obtener la parte real del resultado del paso 4 (necesario debido a imprecisiones computacionales que originan parte imaginaria en este paso).
6. Multiplicar el resultado del paso 5 por $(-1)^{x+y}$, para deshacer el cambio del paso 1.

A $\hat{h}(u, v)$ se le llama filtro porque, al igual que en el caso de señales unidimensionales, suprimirá ciertas frecuencias (espaciales) en la transformada mientras deja pasar otras. Veremos ahora algunos tipos de filtros básicos y sus propiedades.

A.2.2. Eliminación de la componente continua.

Supongamos que queremos forzar el valor medio de una imagen a cero, para eliminar brillos; de la definición de la DFT vemos que

$$\hat{f}(0, 0) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y),$$

con lo cual, el valor medio de una imagen viene dado por $\hat{f}(0, 0)$. Si fijamos este término a cero en el dominio de la frecuencia y tomamos su IDFT, el valor medio de la imagen resultante será cero. Asumiendo que la transformada haya sido centrada como habíamos comentado antes, podemos realizar esta operación multiplicando todos los valores de $\hat{f}(u, v)$ por la función del filtro:

$$\hat{h}(u, v) = \begin{cases} 0 & \text{si } (u, v) = \left(\frac{M}{2}, \frac{N}{2}\right) \\ 1 & \text{resto} \end{cases} \quad (\text{A.1})$$

Lo único que hace el filtro (A.1) es fijar $\hat{f}(0,0)$ a cero y dejar todas las otras componentes frecuenciales intactas, como pretendíamos. La imagen procesada (con media cero) se puede obtener tomando la transformada inversa de $\hat{h}(u, v) \cdot \hat{f}(u, v)$. El resultado de procesar una imagen de esta forma se obtiene en la Figura A.3, aunque los valores negativos resultantes de la imagen se han redondeado a cero, es decir, la imagen mostrada, siendo rigurosos, no tiene media cero. No obstante, lo que sí puede apreciarse es la bajada general de intensidad en la imagen, lo que facilita la inspección (en este caso) de los defectos en un microchip, que quedan resaltados.

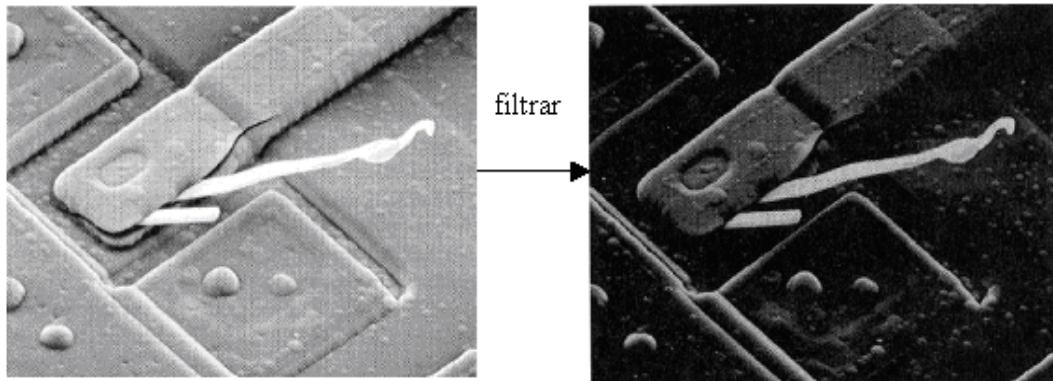


Figura A.3: Eliminación de la componente continua en una imagen. Fuente: [11].

A.2.3. Filtro paso alto

Las bajas frecuencias en la transformada de Fourier son las responsables de la apariencia de nivel de gris general de una imagen sobre áreas suaves, mientras que las altas frecuencias son responsables del detalle, es decir, de aspectos tales como los bordes y el ruido. Podemos ver que lo que hemos hecho en la Figura A.3 es pasar la imagen por un filtro paso alto con mínima frecuencia de corte inferior.

De forma más general podemos emplear un filtro con la siguiente función de transferencia:

$$\hat{h}(u, v) = \begin{cases} 0 & \text{si } D(u, v) < D_0 \\ 1 & \text{si } D(u, v) \geq D_0 \end{cases}$$

Donde D_0 será la frecuencia de corte y $D(u, v)$, la distancia del origen del espectro al vector de frecuencias (u, v) . El filtro expresado de forma gráfica sería el mostrado en la Figura A.4. El color blanco indica valor ‘1’, mientras que el negro se corresponde con ‘0’. Vemos que el número de componentes frecuenciales que no dejamos pasar es mayor que en el caso de la Figura A.3.

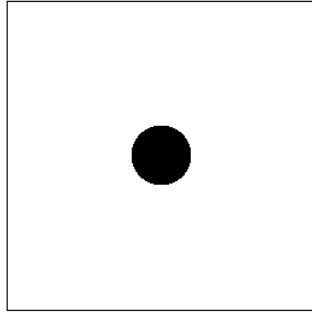


Figura A.4: Función de transferencia de un filtro paso alto ideal.

El resultado de filtrar una imagen con esta máscara se muestra en la Figura A.5. Al igual que antes, vemos que se conservan los cambios abruptos de intensidad, mientras que se reduce la intensidad media de la imagen; pero ahora, adicionalmente, podemos apreciar un efecto conocido como “ringing” debido a que el filtro empleado es ideal. Los filtros ideales tienen una IDFT que está relacionada con la función sinc, la cual, debida a sus lóbulos produce esos ‘bultos’ indeseables en la imagen final.

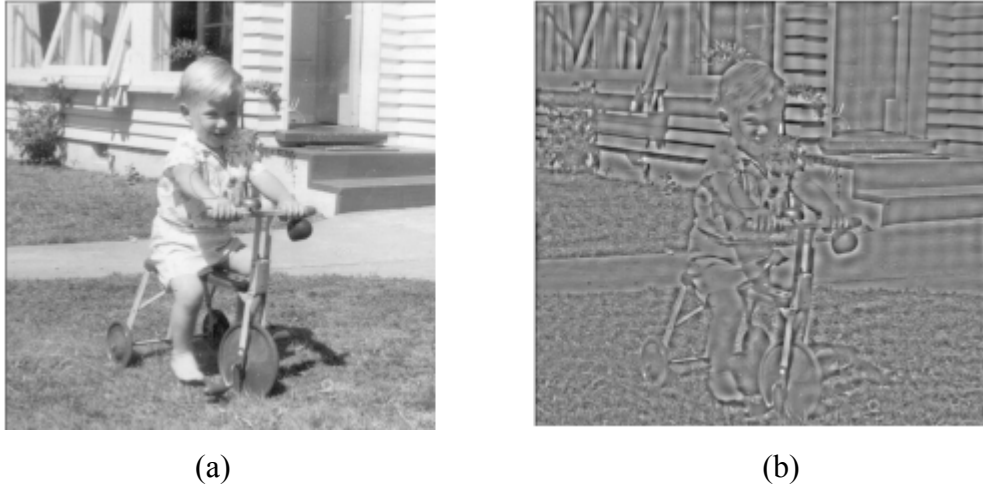


Figura A.5: Resultado de filtrar una imagen por un filtro paso alto ideal. (a): Imagen original. (b): Imagen resultante. Fuente: [35].

El ‘ringing’ se puede eliminar suavizando la caída del filtro, es decir, haciendo que en vez de ser ideal sea de tipo Butterworth, con la siguiente función de transferencia:

$$\hat{h}(u, v) = \frac{1}{1 + [D_0 / D(u, v)]^{2n}}$$

o, alternativamente, de tipo gaussiano, con función de transferencia

$$\hat{h}(u, v) = 1 - \exp(-D^2(u, v) / 2D_0^2)$$

Con lo cual nos queda un filtro más suave, como el de la Figura A.6. En este caso, el resultado de filtrar una imagen se muestra en la Figura A.7. Vemos ahora que el ‘ringing’ se ha reducido. Este resultado es de especial importancia, ya que los filtros de Gabor, al ser de tipo gaussiano, no presentan ‘ringing’.

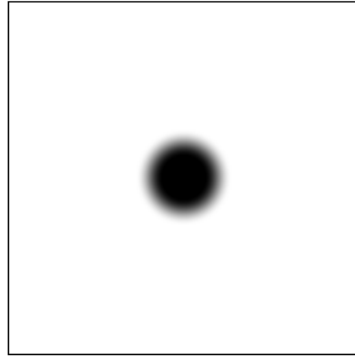


Figura A.6: Función de transferencia de un filtro paso alto gaussiano.



(a)



(b)

Figura A.7: Resultado de filtrar una imagen por un filtro paso alto gaussiano. (a): Imagen original. (b): Imagen resultante. Fuente: [35].

A.2.4. Filtro paso bajo

A continuación presentamos el caso del filtro paso bajo, cuya función de transferencia es:

$$\hat{h}(u, v) = \begin{cases} 1 & \text{si } D(u, v) < D_0 \\ 0 & \text{si } D(u, v) \geq D_0 \end{cases}$$

y cuya máscara está representada en la Figura A.8(a). El propósito de un filtro paso bajo es eliminar los cambios abruptos de intensidad; no obstante, al igual que hicimos con el filtro paso alto, un filtro paso bajo ideal también muestra problemas de ‘ringing’, con lo que, de nuevo, conviene usar su equivalente gaussiano, mostrado en la Figura A.8(b) y con función de transferencia

$$\hat{h}(u, v) = \exp(-D^2(u, v) / 2\sigma^2)$$



Figura A.8: Función de transferencia de filtros paso bajo. (a): Ideal. (b): Gaussiano.

Un ejemplo sencillo se muestra en la Figura A.9. En ella vemos cómo la imagen original (Fig. A.9(a)) se transforma tras pasar por un filtro paso bajo ideal (Fig. A.9(b)) ó por un filtro paso bajo gaussiano (Fig. A.9(c)). Obsérvese que en el caso último no hay efecto de ‘ringing’.

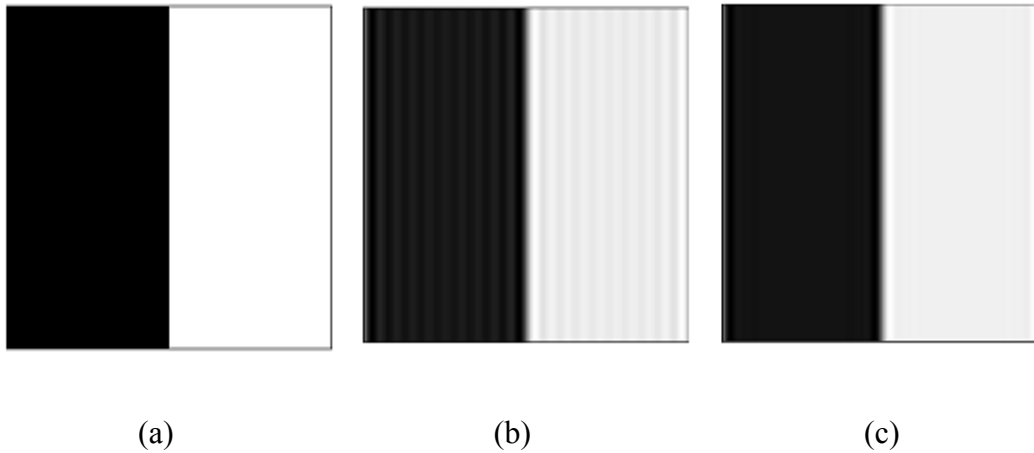


Figura A.9: (a): Imagen original. (b): Resultado de filtrar (a) con un filtro paso bajo ideal. (c): Resultado de filtrar (a) con un filtro paso bajo gaussiano. Fuente: [11].

A.2.5. Filtro de banda rechazada

En la Figura A.10 se muestra la máscara de un filtro de banda rechazada. Como se ve, este filtro deja pasar todas las frecuencias excepto una franja en forma de anillo, que serán todas las frecuencias de un módulo dado y cualquier orientación.

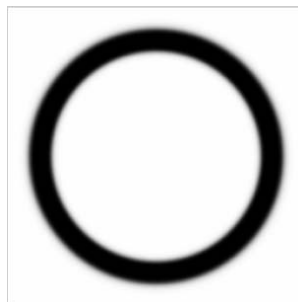


Figura A.10: Función de transferencia de un filtro de banda rechazada

Dada la imagen de la Figura A.11(a), que es una foto corrompida por ruido de tipo sinusoidal superpuesto, podemos filtrarla para mejorar su calidad sin más que emplear un filtro que bloquee la frecuencia específica de dicho ruido, y así obtener la imagen en la Figura A.11(b).



(a)

(b)

Figura A.11: (a): Imagen corrompida por ruido sinusoidal. (b) Resultado de filtrar (a) con un filtro paso banda a la frecuencia de la señal interferente. Fuente: [35].

En realidad, no era necesario que la banda atenuada fuese una circunferencia completa, dado que el ruido era una senoide vertical, y, por tanto, con bloquear los puntos de arriba y abajo de la misma habría sido suficiente. Con el filtro empleado bloquearíamos sinusoides sin importar su orientación.

Apéndice B. Filtros de Gabor unidimensionales

B.1. Generalidades

Un filtro complejo de Gabor en una dimensión se define como el producto de una función gaussiana multiplicado por una senoide compleja

$$g(t) = k \exp(j\theta) w(at) s(t) \quad (\text{B.1})$$

$$\text{donde } w(t) = \exp(-\pi t^2)$$

$$s(t) = \exp(2\pi j f_0 t)$$

$$\exp(j\theta) s(t) = \exp(j(2\pi f_0 t + \theta)) = \cos(2\pi f_0 t + \theta) + j \sin(2\pi f_0 t + \theta) \quad (\text{B.2})$$

Aquí k , θ , f_0 son parámetros del filtro. Podemos pensar en los filtros de Gabor complejos como dos filtros desfasados, uno de ellos situado en la parte real y el otro en la parte imaginaria. La parte real contiene al filtro $g_r(t) = w(t) \cos(2\pi f_0 t + \theta)$ y la parte imaginaria, al filtro $g_i(t) = w(t) \sin(2\pi f_0 t + \theta)$

Tomando la transformada de Fourier de (B.1-2) anterior tenemos

$$\begin{aligned} \hat{g}(f) &= k \exp(j\theta) \int_{-\infty}^{\infty} \exp(-j2\pi ft) w(at) s(t) dt = \\ &= k e^{j\theta} \int_{-\infty}^{\infty} \exp(-j2\pi(f - f_0)t) w(at) dt = \frac{k}{a} \exp(j\theta) \hat{w}\left(\frac{f - f_0}{a}\right) \end{aligned}$$

donde $\hat{w}(f) = w(f) = e^{-\pi f^2}$. Por tanto, la transformada de Fourier de un filtro de Gabor es una función gaussiana centrada en f_0 y con anchura proporcional al parámetro a .

El pico de la respuesta en frecuencia está, por tanto, en f_0 . Para obtener el ancho de banda a mitad de magnitud fijamos $w\left(\frac{f-f_0}{a}\right) = \exp\left(-\pi \frac{f-f_0}{a}\right) = 0.5$.

De ahí obtenemos que la magnitud mitad de pico se consigue para $f - f_0 \pm \sqrt{a^2 \log 2\pi} = 0.4697a \approx 0.5a$.

Vemos entonces que el ancho de banda a magnitud mitad es prácticamente igual a a ; de esta forma a puede ser interpretada como el ancho de banda del filtro a magnitud mitad.

B.2. Eliminación de la componente continua

Dependiendo del valor de f_0 y a el filtro puede tener una componente DC importante, que en la mayoría de los casos no es deseable (no fue así, sin embargo, para el experimento del Capítulo 4). Una forma común de fijar a cero la componente continua es el restarle la salida de un filtro paso bajo gaussiano,

$$h(t) = g(t) - c \cdot w(bt) = k \exp(j\theta) w(at) s(t) - c \cdot w(bt)$$

Por tanto

$$\hat{h}(f) = \hat{g}(f) - \frac{c}{b} \hat{w}\left(\frac{f}{b}\right)$$

Para obtener una respuesta en DC de valor cero necesitamos hacer $\frac{c}{b} \hat{w}(0) = \hat{g}(0)$

$$\text{y } c = b \hat{g}(0) = b \frac{k}{a} \exp(j\theta) \hat{w}\left(\frac{f_0}{a}\right).$$

Donde nos hemos valido del hecho de que $\hat{w}(f) = \hat{w}(-f)$. Por tanto

$$h(t) = g(t) - b \hat{g}(0) = k \exp(j\theta) (w(at) s(t) - \frac{b}{a} \hat{w}\left(\frac{f_0}{a}\right) w(bt))$$

$$\hat{h}(f) = \frac{k}{a} \exp(j\theta) \left(\hat{w}\left(\frac{f-f_0}{a}\right) - \hat{w}\left(\frac{f_0}{a}\right) \hat{w}\left(\frac{f}{b}\right) \right)$$

Es conveniente hacer que $b = a$, en cuyo caso

$$h(t) = k \exp(j\theta) w(at) \left(s(t) - \hat{w}\left(\frac{f_0}{a}\right) \right)$$
$$\hat{h}(f) = \frac{k}{a} e^{j\theta} \left(\hat{w}\left(\frac{f-f_0}{a}\right) - \hat{w}\left(\frac{f_0}{a}\right) \hat{w}\left(\frac{f}{a}\right) \right)$$

Bibliografía

[1] R.O. Duda, P.O. Hart, and D.G. Stork. "Pattern Classification". Wiley, New York, NY, 2001.

[2] L. Breiman. "Arcing classifiers". The annals of Statistics, 1998

[3] G.R. Doddington. "Speaker Recognition – Identifying People by their voices". Proceedings of the IEEE, vol 73, nº. 11, 1985, pp. 1651-1663.

[4] J. Novovicová. "Text Document Classification". Ercim News, 2005
<http://www.ercim.org/>.

[5] T.L. Dimond. "Devices for reading handwritten characters". Proceedings of Eastern Joint Computer Conference, 1957, pp 232-237.

[6] S.Z. Li and A.K. Jain. "Handbook of face recognition". Springer. New York, 2005.

[7] J.D. Woodward. "Biometrics: A Look at Facial Recognition". RAND Public Safety and Justice. Santa Monica, CA, 2003

[8] C.C. Venters, J.P. Eakins, R.J. Hartley: "The User Interface and Content-Based Image Retrieval Systems". BCS-IRSG Annual Colloquium on IR Research, 1997

[9] I. Epifanio (2002): "Descripción de Texturas. Aplicación a su Compresión y Clasificación". Valencia, Universitat de València, 2002. pp. 1-8

-
- [10] R. Kindermann and J. Laurie Snell. "Markov Random Fields and Their Applications". AMS Books Online, 1980.
- [11] R.C. González. "Digital Image Processing". Prentice Hall, 2008.
- [12] A. Bodnarova, M. Bennamoun, K. Kubik. "Suitability analysis of techniques for flaw detection in textiles using texture analysis". *Pattern analysis and applications*, vol. 3, n° 3, 2000, pp. 254-266.
- [13] M. Gric. "Texture-based Image Retrieval in MPEG-7 Multimedia System". EUROCON'2001, Trends in Communications, International Conference on. Zagreb Univ, vol. 2, 2001, pp. 365-368.
- [14] J.R. Movellan: "Tutorial on Gabor Filters". Technical report, MPLab Tutorials, Univ. of California, San Diego, 2005.
- [15] J. Arenas-García, J. Larsen, L. Kai Hansen, A. Meng. "Optimal filtering of dynamics in short-time features for music organization". 7th International Conference on Music Information Retrieval, 2006, pp. 290-295.
- [16] J.G. Daugman. "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *J. Opt. Soc. Am. A* 2, 1985, pp. 1160-1169.
- [17] R. Porter, N. Canagarajah. "Gabor Filters for Rotation Invariant Texture Classification". IEEE International Symposium on Circuits and Systems, vol. 2, 1997, pp. 1193-1196.
- [18] Y.Q. Chen. "Statistical Geometrical Features for Texture Classification". *Pattern Recognition*, vol. 28, n° 4, 1995, pp 537-552.

-
- [19] W.K. Lam. "Classification of Rotated and Scaled Textures by Local Linear Operators". IEEE Int. Symp. on Circuits and Systems,, vol. 1, 1995, pp. 243-246.
- [20] T. Chang and C.J. Kuo. "Texture Analysis and Classification with Tree-Structured Wavelet Transform". IEEE Trans. Image Processing, vol. 2, nº 4, 1993, pp. 429-441.
- [21] P. Brodatz. "Textures: A Photographic Album for Artists and Designers". Dover: New York, 1966.
- [22] Jolliffe I.T. "Principal Component Analysis". Springer Series in Statistics, 2002.
- [23] D. Peña. "Análisis de Datos Multivariantes". Mc Graw Hill, 2003.
- [24] F. van den Berg (consulta hecha en noviembre de 2009). "Introduction to Matlab + Mathematical aspects of bilinear factor models". Disponible en <http://www.models.life.ku.dk>
- [25] "Descomposición en valores singulares". Instituto de Matemáticas Universidad de Sevilla. (consulta hecha en noviembre de 2009). Disponible en <http://ma1.eii.us.es/miembros/cobos/Algoritmos%20AN/Singulares.pdf>.
- [26] P.D. Sampson, et al. "Neurobehavioral effects of prenatal alcohol: Part II. Partial Least Squares analysis". Vol. 11, nº 5, 1989, pp. 477-491.
- [27] J. Arenas-Garcia and G. Camps-Valls. "Efficient Kernel Orthonormalized PLS for Remote Sensing Applications". IEEE Transactions on Geoscience and Remote Sensing. Volume 46, 2008, pp. 2872-2881.
- [28] Image After (consulta hecha en marzo de 2009). Base de datos de texturas. Disponible en <http://www.imageafter.com/>.

[29] Mayang's Free Textures (consulta hecha en marzo de 2009). Base de datos de texturas. Disponible en <http://mayang.com/textures/>.

[30] Vison Texture (consulta hecha en marzo de 2009). Base de datos de texturas. Disponible en:
<http://vismod.media.mit.edu/vismod/imagery/VisionTexture/vistex.html>.

[31] Texture and Color (consulta hecha en marzo de 2009). University of Western Australia. Disponible en http://local.wasp.uwa.edu.au/~pbourke/texture_colour/.

[32] Got3D (consulta hecha en marzo de 2009). 3D Models and Textures. Disponible en <http://www.got3d.com/>.

[33] CG Textures (consulta hecha en marzo de 2009). Base de datos de texturas. Disponible en <http://www.cgtextures.com/>.

[34] A.V. Oppenheim, R.W. Schafer. "Tratamiento de Señales en Tiempo Discreto". Prentice Hall, 2000.

[35] P. Bourke. "Image Filtering in the Frequency Domain", 1998. Disponible en <http://local.wasp.uwa.edu.au/~pbourke/miscellaneous/imagefilter/>